

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им. В.П. АСТАФЬЕВА»  
(КГПУ им. В.П. Астафьева)

Кафедра-разработчик  
*Кафедра информатики и информационных технологий в образовании*

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ**

Направление подготовки:

44.03.05 Педагогическое образование (с двумя профилями подготовки)  
направленность (профиль) образовательной программы  
*«Математика и информатика»*

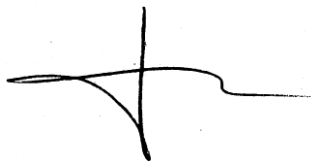
Квалификация (степень) выпускника  
БАКАЛАВР

Красноярск 2019

Рабочая программа дисциплины «Языки и методы программирования» составлена кандидатом педагогических наук, доцентом кафедры ИИТвО Сокольской М.А., кандидатом физико-математических наук, доцентом кафедры ИИТвО Романовым Д.В.,

Рабочая программа дисциплины обсуждена на заседании кафедры ИИТвО

протокол № 10 от 03.05.2017 г.



Заведующий кафедрой \_\_\_\_\_ Пак Н.И

Одобрено научно-методическим советом ИМФИ  
«26» мая 2017 г. Протокол №9

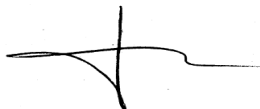


Председатель \_\_\_\_\_ Бортновский С.В.

Рабочая программа дисциплины актуализирована кандидатом педагогических наук, доцентом кафедры ИИТвО Сокольской М.А., кандидатом физико-математических наук, доцентом кафедры ИИТвО Романовым Д.В.

Рабочая программа дисциплины дополнена и скорректирована на заседании кафедры 04.04.2018 г. протокол № 7

Заведующий кафедрой



Пак Н.И.

Одобрено научно-методическим советом ИМФИ  
23.05.2018 протокол № 8

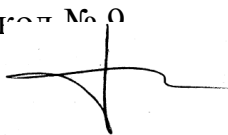
Председатель



Бортновский С.В

Рабочая программа дисциплины актуализирована кандидатом педагогических наук, доцентом кафедры ИИТвО Сокольской М.А., кандидатом физико-математических наук, доцентом кафедры ИИТвО Романовым Д.В.

Рабочая программа дисциплины дополнена и скорректирована на заседании кафедры 08.05.2019 г. протокол № 9



Заведующий кафедрой \_\_\_\_\_ Пак Н.И.

Одобрено научно-методическим советом ИМФИ  
16.05.2019 протокол №8



Председатель \_\_\_\_\_ Бортновский С.В.

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### 1.1. Место дисциплины в структуре образовательной программы

Рабочая программа дисциплины «Языки и методы программирования» для подготовки обучающихся по направлению 44.03.05 «Педагогическое образование (с двумя уровнями подготовки)», профиль «Математика и информатика», «Физика и информатика» разработана в соответствии с требованиями федерального государственного образовательного стандарта высшего образования, утвержденного 09 февраля 2016 г. № 91; профессиональным стандартом «Педагог», зарегистрированным в Минюсте РФ 06.12.2013 №30550, рабочим учебным планом подготовки бакалавров в КГПУ им. В.П. Астафьева по указанному направлению, утвержденном на Ученом совете университета.

Дисциплина «Языки и методы программирования» относится к вариативной части учебного плана подготовки бакалавров по программе «Педагогическое образование», профили «Математика и информатика», «Физика и информатика» и изучается в 3, 4 и 5 семестрах. Код дисциплины в учебном плане – Б1.В.ОД.22.

Рабочая модульная программа предназначена для преподавателей и студентов, являющихся субъектами образовательного процесса в рамках данной дисциплины.

Дисциплина «Языки и методы программирования» опирается на знания и способы деятельности, сформированные в предшествующей дисциплине «Информатика».

### 1.2. Общая трудоемкость дисциплины - в З.Е., часах и неделях

Трудоемкость дисциплины (общий объем времени, отведенного на изучение дисциплины) по очной форме обучения составляет **9 ЗЕТ**, из них:

Аудиторных часов **134**:

Лекций – **64**

Лабораторных работ – **70**

Часов самостоятельной работы – **118**

Дисциплина реализуется на 2 и 3 курсах в 3, 4 и 5 семестрах. Контроль – **72** часа (экзамен – 3 и 5 семестры, зачет – 4 семестр) – **324 часа**

### 1.3 Цель и задачи дисциплины «Языки и методы программирования»

Предметная подготовка студента в педагогическом вузе осуществляется в три этапа:

*Подготовительный этап* - обобщение и систематизация базовых учебных элементов школьного предмета.

*Фундаментальный этап* - глубокое теоретическое обобщение базовых учебных элементов (универсальных учебных действий) школьного предмета.

*Методический этап* – включение фундаментальных предметных знаний в структуру профессиональной деятельности будущего педагога, как средства реализации его учебно-воспитательных функций.

Курс «Языки и методы программирования» является основным звеном подготовки учителя на фундаментальном этапе обучения. Базовые знания, умения и компетенции, формируемые и развиваемые в курсе «Языки и методы программирования», закладывают фундамент предметной и профессиональной подготовки выпускников профиля.

***Целью изучения дисциплины является:***

формирование системы понятий, знаний, умений, навыков и компетенций в области структурного, объектно-ориентированного, функционального и логического программирования на базе языков C, C++, Lisp и Prolog.

Выбор цели обусловлен следующими положениями: во-первых, программирование является существенной частью предметной подготовки педагогов-бакалавров этого профиля. Во-вторых, программирование является предметом, позволяющим не только развивать способы алгоритмической мыслительной деятельности, но и реализовывать возможности межпредметных связей.

**Задачи курса.**

1. Формирование культуры работы в динамичной, быстроразвивающейся области, регулируемой сообществом разработчиков и пользователей продукта.
2. Формирование представлений о структурном программировании на примере языка C и основ C++.
3. Формирование представлений о структурах данных: стеках, очередях, бинарных деревьях, их структуры и назначение.
4. Формирование представлений о технологиях параллельного программирования на примере библиотеки MPI для языков C/C++
5. Формирования знаний и опыта в объектно-ориентированном программировании на примере языка C++.
6. Формирование представлений о логическом и функциональном программировании на примере языков Prolog и Lisp

**1.4. Основные разделы содержания  
3 семестр**

**Входной модуль.**

**Тема В.1** История создания языков программирования. Парадигмы программирования

**Модуль 1. Базовые алгоритмические конструкции в языке C/C++**

**Тема 1.1.** Структура программы в C. Основные библиотеки. Простые типы данных и их характеристики. Операторы языка

**Тема 1.2.** Переменные и константы. Функции ввода-вывода. Формы записи оператора if и его применение. Простые и составные условия, логические операторы. Множественный выбор и его реализация в C.

**Тема 1.3** Синтаксис циклов с параметром, предусловием и постусловием.

**Тема 1.4.** Работа с файлами

## **Модуль 2. Указатели и массивы в С**

**Тема 2.1.** Указатели в языке С. Роль указателей в программировании

**Тема 2.2.** Функции. Передача данных в функцию

**Тема 2.3.** Статические и динамические одномерные и двумерные массивы. Связь массивов и указателей

**Тема 2.4.** Строки как массивы символов

## **Модуль 3. Структуры и списки в С**

**Тема 3.1.** Структуры в языке С. Массивы структур

**Тема 3.2.** Однонаправленные списки.

**Тема 3.3.** Стеки и очереди как разновидности списков с ограничением по дисциплине обслуживания. Моделирование реальных процессов с помощью очередей и стеков

**Тема 3.4.** Двухнаправленные списки (Деки)

**Тема 3.5.** Бинарные деревья.

**Тема 3.6.** Основы анализа эффективности алгоритмов

## **4 семестр**

### ***Модуль 4. Основы параллельного программирования***

**Тема 4.1.** Введение. Понятие параллельной обработки информации. Многопроцессорные вычислительные системы. Классификация. Рейтинг Топ-500.

**Тема 4.2.** Библиотека MPI. Основные функции библиотеки

**Тема 4.3.** Парные операции передачи сообщений. Режимы передачи сообщений

**Тема 4.4.** Коллективные операции передачи сообщений. Функции MPI\_Bcast и MPI\_Reduce.

**Тема 4.5.** Коллективные операции передачи сообщений. Функции MPI\_Scatter (MPI\_Scatterv) и MPI\_Gather (MPI\_Gatherv).

**Тема 4.6.** Решение системы линейных уравнений параллельным методом Гаусса.

### ***Модуль 5. Основы объектно-ориентированного программирования***

**Тема 5.1.** История ООП. Понятие класса и объекта класса. Примеры языков программирования, основанных на ООП

**Тема 5.2.** Инкапсуляция как принцип ООП. Реализация инкапсуляции в С++.

**Тема 5.3.** Наследование как принцип ООП. Проектирование иерархии классов. Реализация наследования в С++. Особенности наследования с разным уровнем доступа (private, protected, public)

**Тема 5.4.** Полиморфизм как принцип ООП. Реализация полиморфизма через переопределение функций базовых классов. Переопределение операторов

**Тема 5.5.** Библиотеки классов в С++. Работа со списками, потоками ввода-вывода, строками.

**Тема 5.6.** Шаблоны классов в С++.

## 5 семестр

### Модуль 6. Основы функционального программирования

**Тема 6.1.** Введение. Принципы функционального программирования. Язык Lisp. Атомы и S-выражения в языке Lisp. Основные функции и особенности записи программ.

**Тема 6.2.** Пользовательские функции в языке Lisp. Именованные и безымянные функции.

**Тема 6.3.** Основные способы определения ветвлений и циклов в языке Lisp.

### Модуль 7. Основы логического программирования

**Тема 7.1.** Введение. Принципы логического программирования. Язык Prolog. Структура программы на языке Prolog. Предикаты. Факты, правила и их описание.

**Тема 7.2.** Списки в языке Prolog и их рекурсивная обработка.

## 1.5. Планируемые результаты обучения

Процесс изучения дисциплины направлен на формирование следующих компетенций:

### а) общекультурные:

**ОК-6:** способность к самоорганизации и самообразованию.

### б) общепрофессиональные:

**ОПК-1:** готовность сознавать социальную значимость своей будущей профессии, обладать мотивацией к осуществлению профессиональной деятельности.

### в) профессиональные:

**ПК-11:** готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования.

Задачи освоения дисциплины	Планируемые результаты обучения по дисциплине (дескрипторы)	Код результата обучения (компетенция)
Формирование культуры работы в динамичной, быстроразвивающейся области, регулируемой сообществом разработчик в и пользователей продукта.	– сформировать представление о языке как знаково-символьной системе записи выполняемых алгоритмов; – носителя информации о технике выполнения требуемой задачи в форме, облегчающей практику одной или нескольких существующих парадигм разработки прикладного ПО; – типичных конфликтах, возникающих вследствие роста сложности разрабатываемого ПО, и методиках их локализации и разрешения; – стандартах и процессах их разработки и внедрения.	<b>ОПК-1</b> готовность сознавать социальную значимость своей будущей профессии, обладать мотивацией к осуществлению профессиональной деятельности  <b>ОК-6:</b> способность к самоорганизации и самообразованию.
	<b>уметь</b> – работать с интерфейсом нескольких сред программирования; – использовать методы отладки программы, предлагаемые средой	



	<p>разработки;</p> <p><b>владеть:</b></p> <ul style="list-style-type: none"> <li>– способами поиска ошибок в программе;</li> <li>– средствами отладки программ;</li> <li>– функционалом выбранной среды программирования</li> </ul>	
<p>Формирование представлений о структурном программировании на примере языка С и основ С++.</p>	<p><b>Знать</b></p> <ul style="list-style-type: none"> <li>– сущность структурного подхода к программированию, его особенности и реализацию с помощью языков С, С++, Lisp;</li> <li>– простые и сложные типы данных и их применение при написании программ;</li> <li>– синтаксис основных конструкций языка, их применение при реализации разработанных алгоритмов</li> <li>– основные приемы работы с оперативной памятью с помощью возможностей языка</li> <li>– назначение и состав базовых библиотек.</li> </ul>	<p><b>ОПК-1</b> готовность сознавать социальную значимость своей будущей профессии, обладать мотивацией к осуществлению профессиональной деятельности</p> <p><b>ПК-11:</b> готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования</p>
	<p><b>Уметь</b></p> <ul style="list-style-type: none"> <li>– разрабатывать линейные, разветвляющиеся и циклические алгоритмы решения учебно-тренировочных задач;</li> <li>– самостоятельно программировать основные компоненты базовых алгоритмов;</li> <li>– использовать специализированное программное обеспечение для программирования;</li> <li>– разрабатывать алгоритмы, использующие возможности языка по работе с оперативной памятью;</li> <li>– осуществлять информационно-поисковую деятельность необходимых ресурсов в сети Интернет;</li> </ul>	
	<p><b>Владеть</b></p> <ul style="list-style-type: none"> <li>– разработкой алгоритмов, реализующих методы структурного программирования;</li> <li>– программированием простых алгоритмов;</li> <li>– программирования сложных алгоритмов с использованием динамических структур;</li> </ul>	
<p>Формирование представлений о структурах данных: стеках, очередях, бинарных деревьях, их структуры и назначение.</p>	<p><b>Знать</b></p> <ul style="list-style-type: none"> <li>– структуру и назначение динамических структур: стеков, очередей, бинарных деревьев на примере языка С++;</li> <li>– области применения и базовые алгоритмы обработки динамических структур</li> </ul> <p><b>Уметь</b></p> <ul style="list-style-type: none"> <li>– разрабатывать алгоритмы с использованием динамических структур данных;</li> </ul>	<p><b>ПК-11:</b> готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования</p> <p><b>ОК-6:</b> способность к самоорганизации и самообразованию.</p>

	<ul style="list-style-type: none"> <li>– распознавать задачи, в которых необходимы динамические структуры</li> </ul>	
	<p><b>Владеть</b></p> <ul style="list-style-type: none"> <li>– разработкой и программированием сложных алгоритмов с использованием динамических структур;</li> </ul>	
Формирование представлений о технологиях параллельного программирования на примере библиотеки MPI для языков C\C++	<p><b>Знать</b></p> <ul style="list-style-type: none"> <li>– место и значение многопроцессорных вычислительных систем для науки и промышленности;</li> <li>– структуру параллельных алгоритмов и их особенности ограничения и преимущества технологий параллельного программирования спектр нерешенных проблем в области параллельных вычислений</li> </ul> <p><b>Уметь</b></p> <ul style="list-style-type: none"> <li>– проектировать алгоритм в технологии MPI программировать спроектированный алгоритм</li> <li>– осознанно использовать виды коллективных и парных операций передачи сообщений в рамках библиотеки mpi языка C/C++</li> </ul> <p><b>Владеть</b></p> <ul style="list-style-type: none"> <li>– разработкой и программированием сложных алгоритмов с использованием технологии параллельного программирования MPI;</li> </ul>	<p><b>ПК-11:</b> готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования</p> <p><b>ОК-6:</b> способность к самоорганизации и самообразованию.</p>
Формирование представлений об объектно-ориентированном программировании на примере языка C++.	<p><b>Знать</b></p> <ul style="list-style-type: none"> <li>– сущность объектно-ориентированного программирования, смысл понятий класса и объекта;</li> <li>– смысл инкапсуляции, наследования и полиморфизма, как основных свойств ООП;</li> <li>– структуру класса и принципы взаимодействия его компонентов;</li> <li>– назначение конструкторов и деструкторов</li> <li>– назначение и синтаксис описания классов-шаблонов;</li> <li>– типизацию методов класса и их назначение</li> </ul> <p><b>Уметь</b></p> <ul style="list-style-type: none"> <li>– проектировать класс по заданной задаче предметной области с учетом ограничений условия задачи;</li> <li>– выделять существенные и несущественные для проектирования класса свойства и методы из условия задачи;</li> <li>– описывать спроектированный класс в программном коде</li> </ul> <p><b>Владеть</b></p> <ul style="list-style-type: none"> <li>– разработкой и программированием сложных алгоритмов на основе принципов объектно-ориентированного программирования;</li> </ul>	<p><b>ПК-11:</b> готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования</p> <p><b>ОК-6:</b> способность к самоорганизации и самообразованию.</p>
Формирование представлений о	<p><b>Знать</b></p> <ul style="list-style-type: none"> <li>– сущность функциональной и</li> </ul>	<p><b>ПК-11:</b> готовность использовать</p>

логическом и функциональном программировании на примере языков Prolog и Lisp	логической парадигм программирования; – суть основных конструкций и типов в языках Lisp и Prolog – процесс проектирования алгоритма в функциональном языке программирования; – процесс проектирования базы знаний в логическом языке программирования; – смысл предикатов и правил в логическом программировании.	систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования  <b>ОК-6:</b> способность к самоорганизации и самообразованию.
	<b>Уметь</b> – записывать алгоритм на функциональном языке программирования; – проектировать систему предикатов для записи базы знаний на логическом языке программирования; – записывать базу знаний с учетом роли фактов и правил.	
	<b>Владеть</b> – способами записи решения задач в функциональном и логическом стилях программирования; – способами обращения к программе, написанной в логическом стиле с вопросом.	

## 1.6. Контроль результатов освоения дисциплины

Текущий контроль осуществляется в ходе выполнения заданий на лабораторных работах, промежуточных тестах по основным модулям программы, а также итогового теста. Итоговая аттестация предусмотрена в виде зачёта/экзамена, критерием получения которого является выполнение текущего контроля и успешное прохождение итогового контрольного мероприятия.

Оценочные средства результатов освоения дисциплины, критерии оценки выполнения заданий представлены в разделе «Фонды оценочных средств для проведения промежуточной аттестации».

## 1.7. Перечень образовательных технологий, используемых при освоении дисциплины

В курсе применяются следующие интерактивные методы и формы проведения учебных занятий: мозговой штурм; сетевая дискуссия, круглый стол в сетевом режиме; совместная экспертиза продуктов деятельности, творческие задания, эвристическая беседа.

Виды учебных действий и формы учебной деятельности в курсе проектируются релевантно образовательным результатам согласно когнитивной таксономии.



## 2. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ ДОКУМЕНТЫ

### 2.1. Технологическая карта обучения дисциплине

Языки и методы программирования

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки),  
направленность (профиль) образовательной программы Математика и информатика

**по очной форме обучения**

Модули. Наименование разделов и тем	Всего часов	Аудиторных часов				Внеауди- торных часов	Формы и методы контроля
		Всего	лекций	Семинаро в	Лаборат. работ		
<b>3 семестр</b>		<b>66</b>	<b>30</b>	<b>-</b>	<b>36</b>	<b>6</b>	
<b>Входной модуль.</b>	<b>4</b>	<b>4</b>	<b>2</b>		<b>2</b>	<b>-</b>	
<b>Тема В.1</b> История создания языков программирования. Парадигмы программирования	4	4	2	-	2	-	<i>Входное тестирование</i>
<b>Модуль 1. Базовые алгоритмические конструкции в языке C/C++</b>	<b>18</b>	<b>16</b>	<b>8</b>	<b>-</b>	<b>8</b>	<b>-</b>	-
<b>Тема 1.1.</b> Структура программы в C. Основные библиотеки. Простые типы данных и их характеристики. Операторы языка	4	4	2	-	2	-	Проверка ЛР№ 1 <i>“Линейные программы”</i>
<b>Тема 1.2.</b> Переменные и константы. Функции ввода-вывода. Формы записи оператора if и его применение. Простые и составные условия, логические операторы. Множественный выбор и его реализация в C.	4	4	2	-	2	-	Проверка ЛР№ 2 <i>“Ветвления”</i>
<b>Тема 1.3</b> Синтаксис циклов с параметром, предусловием и постусловием.	4	4	2	-	2	-	Проверка ЛР№ 3 <i>“Циклы”</i> .
<b>Тема 1.4.</b> Работа с файлами	4	4	2	-	2	-	Проверка ЛР№ 4 <i>“Работа с файлами”</i> .
<i>Промежуточное тестирование по модулю 1</i>	2	-	-	-	-	-	<i>Тестирование по модулю 1</i>
<b>Модуль 2. Указатели и массивы в C</b>	<b>20</b>	<b>16</b>	<b>8</b>	<b>-</b>	<b>8</b>	<b>2</b>	-
<b>Тема 2.1.</b> Указатели в языке C. Роль	4	4	2	-	2	-	Проверка ЛР№ 5

указателей в программировании							“Указатели”.
<b>Тема 2.2.</b> Функции. Передача данных в функцию	4	4	2	-	2	-	Проверка ЛР№ 6 “Функции”.
<b>Тема 2.3.</b> Статические и динамические одномерные и двумерные массивы. Связь массивов и указателей	6	4	2	-	2	2	Проверка ЛР№ 7 “Статические и динамические массивы”.
<b>Тема 2.4.</b> Строки как массивы символов	4	4	2	-	2	-	Проверка ЛР№ 8 “Строки”
<i>Промежуточное тестирование по модулю 2</i>	2	-	-	-	-	-	<i>Тестирование по модулю 2</i>
<b>Модуль 3. Структуры и списки в С</b>	<b>36</b>	<b>30</b>	<b>12</b>		<b>18</b>	<b>4</b>	
<b>Тема 3.1.</b> Структуры в языке С. Массивы структур	4	4	2	-	2	-	Проверка ЛР№ 9 “Структуры”
<b>Тема 3.2.</b> Однонаправленные списки.	8	6	2	-	4	2	Проверка ЛР№ 10 “Однонаправленные списки”
<b>Тема 3.3.</b> Стеки и очереди как разновидности списков с ограничением по дисциплине обслуживания. Моделирование реальных процессов с помощью очередей и стеков	6	6	2	-	4	-	Проверка ЛР№ 11 “Стеки и очереди”
<b>Тема 3.4.</b> Двухнаправленные списки (Деки)	4	4	2	-	2	-	Проверка ЛР№ 12 “Двухнаправленные списки”
<b>Тема 3.5.</b> Бинарные деревья.	8	6	2	-	4	2	Проверка ЛР№ 13 “Бинарные деревья”
<b>Тема 3.6.</b> Основы анализа эффективности алгоритмов	4	4	2	-	2	-	Проверка ЛР№ 14 “Алгоритмы на графах и их эффективность”
<i>Промежуточное тестирование по модулю 3</i>	2	-	-	-	-	-	<i>Тестирование по модулю 3</i>
<b>Экзамен</b>	<b>30</b>						
<b>4 семестр</b>	<b>126</b>	<b>48</b>	<b>24</b>	<b>-</b>	<b>24</b>	<b>78</b>	
<b>Модуль 4. Основы параллельного программирования.</b>	<b>64</b>	<b>24</b>	<b>12</b>	<b>-</b>	<b>12</b>	<b>40</b>	
<b>Тема 4.1.</b> Введение. Понятие параллельной обработки информации. Многопроцессорные	10	4	2	-	2	6	<i>Защита рефератов по развитию и истории</i>

вычислительные системы. Классификация. Рейтинг Топ-500.							<i>параллельных вычислений</i>
<b>Тема 4.2.</b> Библиотека MPI. Основные функции библиотеки	6	4	2	-	2	2	Проверка ЛР№ 15 “Базовые функции библиотеки MPI”
<b>Тема 4.3.</b> Парные операции передачи сообщений. Режимы передачи сообщений	12	4	2	-	2	8	Проверка ЛР№ 16 “Парные операции передачи сообщений”
<b>Тема 4.4.</b> Коллективные операции передачи сообщений. Функции MPI_Bcast и MPI_Reduce	10	4	2	-	2	6	Проверка ЛР№ 17 “Коллективные операции передачи сообщений. Широковещательная рассылка и редукция”
<b>Тема 4.5.</b> Коллективные операции передачи сообщений. Функции MPI_Scatter (MPI_Scatterv) и MPI_Gather (MPI_Gatherv).	10	4	2	-	2	6	Проверка ЛР№ 18 “Коллективные операции передачи сообщений. Распределение данных между процессами”
<b>Тема 4.6.</b> Решение системы линейных уравнений параллельным методом Гаусса.	12	4	2	-	2	8	Проверка ЛР№ 19 “Реализация параллельного алгоритма Гаусса решения СЛУ”
<i>Промежуточное тестирование по модулю 4</i>	4	-	-	-	-	4	<i>Тестирование по модулю 4</i>
<b>Модуль 5. Основы объектно-ориентированного программирования</b>	<b>62</b>	<b>24</b>	<b>12</b>	<b>-</b>	<b>12</b>	<b>38</b>	
<b>Тема 5.1.</b> История ООП. Понятие класса и объекта класса. Примеры языков программирования, основанных на ООП	8	4	2	-	2	4	Проверка ЛР№ 20 “Классы и объекты”
<b>Тема 5.2.</b> Инкапсуляция как принцип ООП. Реализация инкапсуляции в C++.	10	4	2	-	2	6	Проверка ЛР№ 21 “Инкапсуляция. Конструкторы и деструкторы”
<b>Тема 5.3.</b> Наследование как принцип ООП. Проектирование иерархии классов. Реализация наследования в C++. Особенности наследования с разным	10	4	2	-	2	6	Проверка ЛР№ 22 “Наследование. Иерархии классов”

уровнем доступа (private, protected, public)							
<b>Тема 5.4.</b> Полиморфизм как принцип ООП. Реализация полиморфизма через переопределение функций базовых классов. Переопределение операторов	10	4	2	-	2	6	Проверка ЛР№ 23 “Полиморфизм. Перегрузка операторов”
<b>Тема 5.5.</b> Библиотеки классов в C++. Работа со списками, потоками ввода-вывода, строками.	10	4	2	-	2	6	Проверка ЛР№ 24 “Использование библиотек классов в программах”
<b>Тема 5.6.</b> Шаблоны классов в C++.	10	4	2	-	2	6	Проверка ЛР№ 25 “Шаблоны классов”
<i>Промежуточное тестирование по модулю 5</i>	4	-	-	-	-	4	<i>Тестирование по модулю 5.</i>
<b>5 семестр</b>	<b>90</b>	<b>20</b>	<b>10</b>	<b>-</b>	<b>10</b>	<b>34</b>	
<b>Модуль 6. Основы функционального программирования</b>	<b>32</b>	<b>12</b>	<b>6</b>	<b>-</b>	<b>6</b>	<b>20</b>	
<b>Тема 6.1.</b> Введение. Принципы функционального программирования. Язык Lisp. Атомы и S-выражения в языке Lisp. Основные функции и особенности записи программ	8	4	2	-	2	4	Проверка ЛР№ 26 “Простые программы на языке Lisp”
<b>Тема 6.2.</b> Пользовательские функции в языке Lisp. Именованные и безымянные функции.	8	4	2	-	2	4	Проверка ЛР№ 27 “Функции в языке Lisp”
<b>Тема 6.3.</b> Основные способы определения ветвлений и циклов в языке Lisp.	12	4	2	-	2	8	Проверка ЛР№ 28 “Ветвления и циклы на языке Lisp”
<i>Промежуточное тестирование по модулю 6</i>	4	-	-	-	-	4	<i>Тестирование по модулю 6</i>
<b>Модуль 7. Основы логического программирования</b>	<b>22</b>	<b>8</b>	<b>4</b>	<b>-</b>	<b>4</b>	<b>14</b>	
<b>Тема 7.1.</b> Введение. Принципы логического программирования. Язык Prolog. Структура программы на языке Prolog. Предикаты. Факты, правила и их описание.	8	4	2	-	2	4	Проверка ЛР№ 29 “Простые программы на языке Prolog”
<b>Тема 7.2.</b> Списки в языке Prolog и их рекурсивная обработка.	10	4	2	-	2	6	Проверка ЛР№ 30 “Работа со списками на языке Prolog”
<i>Промежуточное тестирование по модулю 7</i>	4	-	-	-	-	4	<i>Тестирование по модулю 7</i>



<b>Итоговый модуль</b>	<b>36</b>	-	-	-	-	-	
Экзамен	36	-	-	-	-	-	
<b>ИТОГО</b>	<b>324</b>	<b>134</b>	<b>64</b>	-	<b>70</b>	<b>118</b>	



## **2.2 Содержание основных разделов и тем дисциплины**

### **Входной модуль**

Понятие языка программирования. Машина Бэббиджа. Низкоуровневые языки. Алгол, Фортран, С, Паскаль, С++, Java. Классификации языков программирования. Понятия семантики, алфавита, синтаксиса языка. Язык нормальных форм Бэкуса. Синтаксические диаграммы Вирта. Сущность и свойства структурного программирования. История создания языков С и С++. Блок-схемы основных алгоритмических конструкций.

#### **Модуль 1. Базовые алгоритмические конструкции в языке С/С++**

##### **Раздел 1. Структура программы в С. Основные библиотеки. Простые типы данных и их характеристики. Операторы языка**

Структура программы в С. Основные библиотеки языка. Подключение библиотек. Основной функционал и назначение базовых библиотек. Простые типы данных и их характеристики. Операторы языка. Приоритет операторов.

##### **Раздел 2. Переменные и константы. Функции ввода-вывода. Формы записи оператора if и его применение. Простые и составные условия, логические операторы. Множественный выбор и его реализация в С.**

Переменные и константы. Инициализация переменных и констант. Функции ввода-вывода. Спецификаторы и модификаторы ввода-вывода. Организация ветвлений. Формы записи оператора if и его применение. Простые и составные условия, логические операторы. Множественный выбор и его реализация в С.

##### **Раздел 3 Синтаксис циклов с параметром, предусловием и постусловием.**

Синтаксис циклов с параметром, предусловием и постусловием. Применение и взаимозаменяемость всех видов циклов. Особенности использования цикла с параметром в С.

##### **Раздел 4. Работа с файлами**

Этапы работы с файлами в С. Основные функции работы с файлами. Модификаторы доступа к файлу и открытия файла. Понятие файлового потока.

### **Модуль 2. Указатели и массивы в С**

#### **Раздел 1. Указатели в языке С. Роль указателей в программировании**

Адресация переменных в памяти компьютера. Понятие указателя. Операторы взятия адреса переменной и разыменования указателя. Передача параметров в функцию через указатель. Отличия от передачи параметра по значению.

#### **Раздел 2. Массивы**

Статические одномерные и двумерные массивы. Связь массивов и указателей. Динамические одномерные и двумерные массивы. Арифметика указателей, особенности сложения и вычитания числа с указателем. Ограничения на выполнение арифметических действий над указателями.

#### **Раздел 3. Строки как массивы символов**

Строки с нуль-символом как признак конца строки. Строка как массив символов. Библиотека обработки строк. Функции обработки строк и особенности их применения.

### **Модуль 3. Структуры и списки в С**

#### **Раздел 1. Структуры в языке С. Массивы структур**

Понятие структуры. Назначение структур. Синтаксис описания структур и объявления соответствующих переменных. Обращение к полям структуры в случае статической и динамической переменных. Массивы структур и особенности их обработки. Запись структур в файл и чтение из файла.

#### **Раздел 2. Списки.**

Списки и их назначение для моделирования реальных процессов. Классификация списков по количеству связей, дисциплине обслуживания. Стеки и очереди как разновидности списков с ограничением по дисциплине обслуживания. Моделирование реальных процессов с помощью очередей и стеков. Двухнаправленные списки (Деки). Бинарные деревья. Значение бинарных деревьев в моделировании

#### **Раздел 3. Основы анализа эффективности алгоритмов**

Анализ эффективности алгоритмов как раздел теории алгоритмов. Понятие трудоемкости алгоритма. Способы расчета трудоемкости алгоритма по количеству действий и по времени работы. Классификация алгоритмов.

**Экзамен 3 семестр**

### ***Модуль 4. Основы параллельного программирования***

#### **Раздел 1. Понятие параллельных вычислений.**

Понятие параллельной обработки информации. Параллельная, многозадачная и распределенная обработка информации. Многопроцессорные вычислительные системы. Классификация МВС. Рейтинг Топ-500. Применение параллельных вычислений в науке и производстве.

#### **Раздел 2. Библиотека MPI.**

Основные функции библиотеки. Парные операции передачи сообщений. Режимы передачи сообщений. Коллективные операции передачи сообщений. Функции MPI\_Bcast и MPI\_Reduce. Время работы параллельной программы. Зависимость времени работы от объема данных и числа процессов. Расчет достижимого ускорения. Коллективные операции передачи сообщений. Функции MPI\_Scatter (MPI\_Scatterv) и MPI\_Gather (MPI\_Gatherv). Равное и произвольное разделение исходных данных между процессами. Решение системы линейных уравнений параллельным методом Гаусса.

### ***Модуль 5. Основы объектно-ориентированного программирования***

#### **Раздел 1. Введение в ООП.**

История ООП. Понятие класса и объекта класса. Примеры языков программирования, основанных на ООП. Инкапсуляция как принцип ООП. Реализация инкапсуляции в С++. Наследование как принцип ООП. Проектирование иерархии классов. Реализация наследования в С++. Особенности наследования с разным уровнем доступа (`private`, `protected`, `public`). Полиморфизм как принцип ООП. Реализация полиморфизма через переопределение функций базовых классов. Переопределение операторов

### **Раздел 2. Дополнительные возможности ООП.**

Библиотеки классов в С++. Работа со списками, потоками ввода-вывода, строками. Шаблоны классов в С++. Назначение шаблонов классов и функций. Синтаксис описания шаблонов. Динамическое определение типа объекта класса-шаблона.

**Зачёт 4 семестр**

## **5 семестр**

### ***Модуль 6. Основы функционального программирования***

#### **Раздел 1. Введение.**

Принципы функционального программирования. Язык Lisp. Атомы и S-выражения в языке Lisp. Основные функции и особенности записи программ.

#### **Раздел 2. Основные синтаксические конструкции языка Lisp.**

Пользовательские функции в языке Lisp. Именованные и безымянные функции. Основные способы определения ветвлений и циклов в языке Lisp. Функция `COND`, циклы `DOLIST`, `DOTIMES`, `DO`. Конструкция `LOOP` и особенности ее синтаксиса. Ключевые слова конструкции `LOOP`.

### ***Модуль 7. Основы логического программирования***

#### **Раздел 1. Введение.**

Принципы логического программирования. Язык Prolog. Структура программы на языке Prolog. Предикаты. Факты, правила и их описание. Поиск с возвратом. Хвостовая рекурсия. Откат с петлями. Списки в языке Prolog и их рекурсивная обработка.

**Экзамен 5 семестр**

## **2.3.Методические рекомендации по освоению дисциплины для обучающихся образовательной программы**

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность (профиль) образовательной программы

Математика и информатика  
**по очной форме обучения**

### **Введение**

Методические рекомендации содержат:

1. Рекомендации по организации работы студента на лекциях и практических занятиях
2. Рекомендации по организации самостоятельной работы студента
3. Рекомендации по работе в модульно-рейтинговой системе.
4. Советы по подготовке к экзаменам.

### **Методические рекомендации по организации работы студента на лекциях**

Во время лекций по дисциплине студент должен уметь сконцентрировать внимание на рассматриваемых проблемах и включить в работу все виды памяти: словесную, образную и моторно-двигательную. Для этого ему необходимо конспектировать материал, излагаемый преподавателем. Во время конспектирования в работу включается моторно-двигательная память, позволяющая эффективно усвоить лекционный материал. Каждому студенту необходимо помнить о том, что конспектирование лекции – это не диктант. Студент должен уметь выделять главное и фиксировать основные моменты «своими словами». Это гораздо более эффективно, чем запись «под диктовку».

На каждой лекции периодически проводится устный опрос студентов по материалам лекций. Подборка вопросов для опроса осуществляется на основе изученного теоретического материала. Такой подход позволяет не только контролировать уровень усвоения теоретического материала, но и организовать эффективный контроль посещаемости занятий на потоковых лекциях.

### **Методические рекомендации по организации работы студента на лабораторных занятиях**

Наряду с прослушиванием лекций по курсу важное место в учебном процессе занимают лабораторные занятия, призванные закреплять полученные студентами теоретические знания.

Перед лабораторным занятием студенту необходимо восстановить в памяти теоретический материал по теме лабораторного занятия. Для этого следует обратиться к конспекту лекций и справочному материалу.

Каждое занятие начинается с повторения теоретического материала по предыдущим темам. После такой проверки студентам предлагается выполнить задачи по программированию по текущей тематике. Задачи носят характер учебно-тренировочных заданий по программированию и разделены на 10 вариантов. В части лабораторных работ присутствует общая для всего курса задача, нацеленная на закрепление ключевых для темы знаний и умений.

Порядок решения задач студентами может быть различным. Преподаватель может установить такой порядок, согласно которому каждый студент в отдельности самостоятельно решает задачу без обращения к каким – либо материалам или к преподавателю. Может быть использован и такой порядок решения задачи, когда предусматривается самостоятельное решение каждым студентом поставленной задачи с использованием конспектов, учебников и других методических и справочных материалов. При этом преподаватель обходит студентов, наблюдая за ходом решения и давая индивидуальные указания. По мере решения задач исходные коды программ демонстрируются преподавателю, корректируются или дорабатываются согласно указанию или запросу преподавателя и прикрепляются в электронный курс.

В конце занятия преподаватель подводит его итоги, даёт оценку активности студентов и уровня их знаний.

Каждому студенту необходимо основательно закреплять полученные знания и вырабатывать навыки самостоятельной работы. С этой целью задачи лабораторных работ сформулированы так, что большинство студентов должны завершать их решение в домашних условиях..

### **Методические рекомендации по организации самостоятельной работы студента**

Для эффективного достижения указанных во **введении рабочей программы** целей обучения по дисциплине процесс изучения материала курса предполагает достаточно интенсивную работу не только на лекциях и лабораторных работах, но и дома в ходе самостоятельной работы.

Поэтому рассмотрим процесс организации самостоятельной внеаудиторной работы студентов. Внеаудиторная самостоятельная работа включает выполнение **заданий лабораторных работ** по каждой теме курса в полном объеме (задания представлены в разделе «**Фонд оценочных средств**» РПД

### **Рекомендации по работе в модульно-рейтинговой системе**

Результаты учебной деятельности студентов оцениваются рейтинговыми баллами. В каждом модуле определяется минимальное и максимальное количество баллов.

Виды деятельности, учитываемые в рейтинге и их оценка в баллах представлена в **Технологической карте дисциплины**, которая входит в состав данного РПД.

Сумма максимальных баллов по всем темам (100) равняется 100%-ному усвоению материала.

Минимальное количество баллов в каждом семестре является обязательным и не может быть заменено никакой другой деятельностью. Дисциплинарный модуль считается изученным, если студент набрал количество баллов в рамках установленного диапазона. Баллы учитываются по каждому семестру отдельно в связи с тем, что курс весьма объёмный и имеет несколько промежуточных точек контроля (экзамен, зачёт, экзамен)

Для получения положительной оценки необходимо набрать не менее 60 баллов, предусмотренных по дисциплине (при условии набора всех обязательных минимальных баллов).

Перевод баллов в академическую оценку осуществляется по следующей схеме: оценка «удовлетворительно» 50 – 72 % баллов, «хорошо» 73 – 86 % баллов, «отлично» 87 – 100 % баллов

#### **Соответствие рейтинговых баллов и академической оценки**

Общее количество набранных баллов	Академическая оценка
50 – 72	3 (удовлетворительно)
73 – 86	4 (хорошо)
87 – 100	5 (отлично)

Дополнительный модуль - необязательный. Количество баллов по дополнительному модулю не включается в общую максимальную сумму баллов, распределяемых по модулям. Работа над проектом – возможность поднять свой рейтинг.

Преподаватель имеет право по своему усмотрению добавлять студенту определенное количество баллов (но не более 5 % от общего количества), в каждом дисциплинарном модуле:

- за активность на занятиях;
- за выступление с докладом на научной конференции;
- за научную публикацию;
- за иные учебные или научные достижения.

#### **Работа с неуспевающими студентами**



Студент, не набравший минимального количества баллов по текущей и промежуточной аттестациям в пределах первого базового модуля, допускается к изучению следующего базового модуля. Ему предоставляется возможность добора баллов в течение двух последующих недель (следующих за промежуточным рейтинг-контролем (тестированием по модулю)) на ликвидацию задолженностей.

Студентам, которые не смогли набрать промежуточный рейтинг или рейтинг по дисциплине в общеустановленные сроки по болезни или по другим уважительным причинам (документально подтвержденным соответствующим учреждением), декан факультета устанавливает индивидуальные сроки сдачи.

Если после этого срока задолженность по неуважительным причинам сохраняется, то назначается комиссия по приему академических задолженностей с обязательным участием заведующего кафедрой и декана (его заместителя). По решению комиссии неуспевающие студенты по представлению декана отчисляются приказом ректора из университета за невыполнение учебного графика.

В особых случаях декан имеет право установить другие сроки ликвидации студентами академических задолженностей.

Неявка студента на итоговый или промежуточный рейтинг-контроль отмечается в рейтинг-листе записью "не явился". Если неявка произошла по уважительной причине (подтверждена документально), деканат имеет право разрешить прохождение рейтинг-контроля в другие сроки. При неуважительной причине неявки в статистических данных деканата проставляется "0" баллов, и студент считается задолжником по данной дисциплине.

Рейтинговая система оценки качества учебной работы распространяется и на студентов, переведенных на индивидуальное обучение.

Если студент желает повысить рейтинг по дисциплине после итогового контроля, то он должен заявить об этом в деканате. Дополнительная проверка знаний осуществляется преподавателем по направлению деканата в течение недели после итогового контроля. При этом преподаватель должен ориентироваться на те темы дисциплины, по которым студент набрал наименьшее количество баллов. Полученные баллы вносятся в единую ведомость оценки успеваемости студентов (в дополнительный модуль) и учитываются при определении рейтинговой оценки в целом по дисциплине. Если студент во время дополнительной проверки знаний не смог повысить рейтинговую оценку, то ему сохраняется количество баллов, набранных ранее.

### 3. КОМПОНЕНТЫ МОНИТОРИНГА УЧЕБНЫХ ДОСТИЖЕНИЙ СТУДЕНТОВ

#### 3.1. Технологическая карта рейтинга дисциплины

Наименование дисциплины	Направление подготовки и уровень образования (бакалавриат, магистратура, аспирантура) Наименование программы/ профиля	Количество з.е.
Языки и методы программирования	<b>Направление подготовки:</b> 44.03.05 «Педагогическое образование» Квалификация (степень): Бакалавр Профиль «Математика и информатика» по <b>очной</b> форме обучения	<b>9</b>
<i>Смежные дисциплины по учебному плану:</i> «Геометрия», «Математический анализ и элементы теории функций», «Математическая логика», «Алгебра», «Теоретические основы информатики», «Философия».		
<i>Предшествующие:</i> «Математика», «Информатика», «Информационные технологии в образовании», «Информационная культура и технологии в образовании»		
<i>Последующие:</i> все последующие дисциплины профиля		

<b>ВХОДНОЙ МОДУЛЬ</b>			
Содержание	Форма работы	Количество баллов 6%	
		min	max
Текущая работа	<i>Входное тестирование</i>	3,6	6
<b>Итого</b>		<b>3,6</b>	<b>6</b>

<b>БАЗОВЫЙ МОДУЛЬ № 1.</b>			
<b>Базовые алгоритмические конструкции в языке C/C++</b>			
Содержание	Форма работы	Количество баллов 17%	
		min	max
Текущая работа	Проверка лабораторной работы № 1 “ <i>Линейные программы</i> ”	1,5	3
Текущая работа	Проверка лабораторной работы № 2 “ <i>Ветвления</i> ”	1,5	3
Текущая работа	Проверка лабораторной	1,5	3

	работы № 3 “Циклы”.		
Текущая работа	Проверка лабораторной работы № 4 “Работа с файлами”.	1,5	3
Промежуточный рейтинг-контроль	<i>Промежуточное тестирование по модулю 1</i>	2,5	5
Итого		<b>8,5</b>	<b>17</b>

<b>БАЗОВЫЙ МОДУЛЬ № 2. Указатели и массивы в С</b>			
Содержание	Форма работы	Количество баллов 17%	
		min	max
Текущая работа	Проверка лабораторной работы № 5 “Указатели”.	1,5	3
Текущая работа	Проверка лабораторной работы № 6 “Функции”.	1,5	3
Текущая работа	Проверка лабораторной работы № 7 “Статические и динамические массивы”.	1,5	3
Текущая работа	Проверка лабораторной работы № 8 “Строки”	1,5	3
Промежуточный рейтинг-контроль	<i>Промежуточное тестирование по модулю 2</i>	2,5	5
Итого		<b>8,5</b>	<b>17</b>

<b>БАЗОВЫЙ МОДУЛЬ № 3. Структуры и списки в С</b>			
Содержание	Форма работы	Количество баллов 30%	
		min	max
Текущая работа	Проверка лабораторной работы № 9 “Структуры”	2	4
Текущая работа	Проверка лабораторной работы № 10 “Однонаправленные списки”	2	4

Текущая работа	Проверка лабораторной работы № 11 “Стеки и очереди”	2	4
	Проверка лабораторной работы № 12 “Двунаправленные списки”	2	4
	Проверка лабораторной работы № 13 “Бинарные деревья”	2	4
	Проверка лабораторной работы № 14 “Алгоритмы на графах и их эффективность”	2	4
Промежуточный рейтинг-контроль	Промежуточное тестирование по модулю 3	3	6
<b>Итого</b>		<b>15</b>	<b>30</b>

<b>Итоговый модуль за 3 семестр</b>			
Содержание	Форма работы	Количество баллов 30%	
		min	max
	<i>Экзамен</i>	18	30
<b>Итого</b>		<b>18</b>	<b>30</b>

<b>БАЗОВЫЙ МОДУЛЬ № 4.</b>			
<i>Основы параллельного программирования.</i>			
Содержание	Форма работы	Количество баллов 40%	
		min	max
Текущая работа	<i>Защита рефератов по развитию и истории параллельных вычислений</i>	3	6
Текущая работа	Проверка лабораторной работы № 15 “Базовые функции библиотеки MPI”	3	6
Текущая работа	Проверка лабораторной работы № 16 “Парные операции передачи сообщений”	3	6
Текущая работа	Проверка лабораторной	3	6

	работы № 17 “Коллективные операции передачи сообщений. Широковещательная рассылка и редукция”		
Текущая работа	Проверка лабораторной работы № 18 “Коллективные операции передачи сообщений. Распределение данных между процессами”	3	6
Текущая работа	Проверка лабораторной работы № 19 “Реализация параллельного алгоритма Гаусса решения СЛУ”	3	6
Промежуточный рейтинг-контроль	Промежуточное тестирование по модулю 4	2	4
Итого		<b>20</b>	<b>40</b>

<b>БАЗОВЫЙ МОДУЛЬ № 5.</b>			
<i>Основы объектно-ориентированного программирования.</i>			
Содержание	Форма работы	Количество баллов 40%	
		min	max
Текущая работа	Проверка лабораторной работы № 20 “Классы и объекты”	3	6
Текущая работа	Проверка лабораторной работы № 21 “Инкапсуляция. Конструкторы и деструкторы”	3	6
Текущая работа	Проверка лабораторной работы № 22 “Наследование. Иерархии классов”	3	6
Текущая работа	Проверка лабораторной работы № 23 “Полиморфизм. Перегрузка операторов”	3	6

Текущая работа	Проверка лабораторной работы № 24 <i>“Использование библиотек классов в программах”</i>	3	6
Текущая работа	Проверка лабораторной работы № 25 <i>“Шаблоны классов”</i>	3	6
Промежуточный рейтинг-контроль	<i>Промежуточное тестирование по модулю 5</i>	2	4
Итого		<b>20</b>	<b>40</b>

<b>Итоговый модуль 4 семестр</b>			
Содержание	Форма работы	Количество баллов 20%	
		min	max
	<i>Зачёт</i>	10	20
Итого		<b>10</b>	<b>20</b>

<b>БАЗОВЫЙ МОДУЛЬ № 6.</b> <i>Основы функционального программирования.</i>			
Содержание	Форма работы	Количество баллов 30%	
		min	max
Текущая работа	Проверка лабораторной работы № 26 <i>“Простые программы на языке Lisp”</i>	4	8
Текущая работа	Проверка лабораторной работы № 27 <i>“Функции в языке Lisp”</i>	4	8
Текущая работа	Проверка лабораторной работы № 28 <i>“Ветвления и циклы на языке Lisp”</i>	4	8
Промежуточный рейтинг-контроль	<i>Промежуточное тестирование по модулю 6</i>	3	6
Итого		<b>15</b>	<b>30</b>

<b>БАЗОВЫЙ МОДУЛЬ № 7.</b> <i>Основы логического программирования.</i>			
---	--	--	--

Содержание	Форма работы	Количество баллов 30%	
		min	max
Текущая работа	Проверка лабораторной работы № 29 “Простые программы на языке Prolog”	6	11
Текущая работа	Проверка лабораторной работы № 30 “Работа со списками на языке Prolog”	6	11
Промежуточный рейтинг-контроль	Промежуточное тестирование по модулю 7	4	8
<b>Итого</b>		<b>16</b>	<b>30</b>

<b>Итоговый модуль 5 семестр</b>			
Содержание	Форма работы	Количество баллов 40%	
		min	max
	Экзамен	20	40
<b>Итого</b>		<b>20</b>	<b>40</b>

<b>ДОПОЛНИТЕЛЬНЫЙ МОДУЛЬ</b>			
Форма работы		Количество баллов	
		min	max
<i>Написание реферата по выбранной теме</i>		6	10
<b>Итого</b>		<b>6</b>	<b>10</b>
Общее количество баллов по дисциплине (по итогам изучения всех модулей, без учета дополнительного модуля)		min	max
		<b>150</b>	<b>300</b>


**Соответствие рейтинговых баллов и академической оценки:**

Общее количество набранных баллов*	Академическая оценка
60–72	3 (удовлетворительно)
73–86	4 (хорошо)
87–100	5 (отлично)

### 3.2. Фонд оценочных средств (контрольно-измерительные материалы)

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Красноярский государственный педагогический университет им. В.П.  
Астафьева»**  
Институт математики, физики и информатики  
(наименование института/факультета)  
Кафедра-разработчик Информатики и информационных технологий в  
образовании  
(наименование кафедры)

УТВЕРЖДЕНО  
на заседании кафедры Протокол № 1  
от «31» августа 2018 г.



ОДОБРЕНО  
на заседании научно-  
методического совета направления  
подготовки Протокол № 2  
от «14» сентября 2018 г.



**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
для проведения текущего контроля и промежуточной аттестации обучающихся  
**«Языки и методы программирования»**  
(наименование дисциплины/модуля/вида практики)

Направление подготовки: 44.03.05 Педагогическое образование (с двумя  
профилями подготовки)

Направленность (профиль) образовательной программы  
Математика и информатика

Квалификация: бакалавр  
по **очной** форме обучения

Составители:

*канд.пед.наук, доцент кафедры ИИТвО Сокольская М.А.*  
*к.ф.-м.н., доцент кафедры ИИТвО Романов Д.В.*



## ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ НА ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Представленный фонд оценочных средств для текущей и промежуточной аттестации соответствует требованиям ФГОС ВО и профессиональным стандартам Педагог (профессиональная деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель), утвержденным приказом Минтруда России от 18.10.2013 N 544н.

Предлагаемые формы и средства аттестации адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 44.03.01 Педагогическое образование, направленность (профиль) образовательной программы: «Математика», квалификация (степень): бакалавр.


Оценочные средства и критерии оценивания представлены в полном объеме. Формы оценочных средств, включенных в представленный фонд, отвечают основным принципам формирования ФОС, установленных в Положении о формировании фонда оценочных средств для текущего контроля успеваемости, промежуточной и итоговой (государственной итоговой) аттестации обучающихся по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры, программам подготовки научно-педагогических кадров в аспирантуре - в федеральном государственном бюджетном образовательном учреждении высшего образования «Красноярский государственный педагогический университет им. В.П. Астафьева», утвержденного приказом ректора № 297 (п) от 28.04.2018.

**Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки по указанной программе.**

Эксперт

учитель информатики высшей категории,  
заместитель директора по учебно-воспитательной работе  
МБОУ «СОШ № 10 с углубленным изучением отдельных  
предметов имени академика Ю.А. Овчинникова»  
г. Красноярск



 Г.С. Карпенко

## **1. Назначение фонда оценочных средств**

1.1. **Целью** создания ФОС дисциплины «Языки и методы программирования» является установление соответствия учебных достижений запланированным результатам обучения и требованиям основной профессиональной образовательной программы, рабочей программы дисциплины.

1.2. ФОС по дисциплине решает **задачи**:

1. Управление процессом приобретения обучающимися необходимых знаний, умений, навыков и формирования компетенций, определенных в образовательных стандартах по соответствующему направлению подготовки.

2. Оценка достижений обучающихся в процессе изучения дисциплины с определением положительных/отрицательных результатов и планирование предупреждающих/корректирующих мероприятий.

3. Обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности через совершенствование традиционных и внедрение инновационных методов обучения в образовательный процесс.

4. Совершенствование процессов самоподготовки и самоконтроля обучающихся.

1.3. ФОС разработан на основании нормативных **документов**:

- федерального государственного образовательного стандарта высшего образования по направлению подготовки 44.03.05 «Педагогическое образование» Квалификация (степень) «Бакалавр»

- положения о формировании фонда оценочных средств для текущего контроля успеваемости, промежуточной и итоговой (государственной итоговой) аттестации обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, программам подготовки научно-педагогических кадров в аспирантуре – в федеральном государственном бюджетном образовательном учреждении высшего образования «Красноярский государственный педагогический университет им. В.П. Астафьева» утвержденного приказом ректора № 297 (п) от 28.04.2018.

**2. Перечень компетенций с указанием этапов их формирования в процессе изучения дисциплины/модуля/прохождения практики**

2.1. **Перечень компетенций**, формируемых в процессе изучения дисциплины:

*а) общекультурные:*

**ОК-6:** способность к самоорганизации и самообразованию.

*б) общепрофессиональные:*

**ОПК-1:** готовность сознавать социальную значимость своей будущей профессии, обладать мотивацией к осуществлению профессиональной деятельности.

*в) профессиональные:*

**ПК-11:** готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования.



## 2.2. Оценочные средства

Компетенция	Дисциплины, практики, участвующие в формировании данной компетенции	Тип контроля	Оценочное средство/КИМ	
			Номер	Форма
ОК-6: способность к самоорганизации и самообразованию.	Общекультурные основы профессиональной деятельности, Иностранный язык, Русский язык и культура речи, Экономика образования, Информационная культура и технологии в образовании, Социология, Физическая культура и спорт, Модуль "Теоретические основы профессиональной деятельности", Педагогика, Модуль "Научные основы учебно- профессиональной деятельности", Основы учебной деятельности студента, Основы научной деятельности студента, Основы математической обработки информации, Физика, Информатика, Теория вероятностей и математическая статистика, Математическая логика, Алгебра, Элементарная математика, Численные методы, Информационные системы и сети, Информационные и коммуникационные технологии в образовании, Архитектура профессионального компьютера и операционные системы, Элективная дисциплина по общей физической подготовке, Элективная дисциплина по подвижным и спортивным играм, Элективная дисциплина по физической культуре для обучающихся с ОВЗ и Инвалидов, Профильное исследование в области информатики, Элементарная алгебра, Элементы алгебры, Элементарная геометрия, Элементы геометрии, Дискретная математика, Избранные вопросы дискретной математики, Компьютерное моделирование, Моделирование информационных систем, Исследование операций, Методы оптимизации, Защита информации, Информационная безопасность, Организация исследовательской деятельности школьников, Intel - обучение для будущего, Дополнительные главы математического анализа, История математики, История математического образования в России, Числовые системы, Дополнительные главы алгебры, Открытые программные средства в школьном курсе информатики, Свободное программное обеспечение в обучении, Инновационные процессы в профильном образовании, Новые качество и методы обучения математике, Профессиональная деятельность учителя информатики, Теория и методика профильного обучения информатике, Классное руководство, Основы классного руководства, Методика обучения математике, Методика обучения информатике, ио первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, Производственная практика, Практика по получению профессиональных умений и опыта профессиональной деятельности, Преддипломная практика, Педагогическая практика интерна, Подготовка к сдаче и сдача государственного экзамена, Подготовка к защите и защита выпускной квалификационной работы	Текущий контроль успеваемости  Промежуточная аттестация	все	Выполнение контрольных работ Экзамен, зачёт
ОПК-1: готовность сознать социальную значимость своей будущей профессии, обладать мотивацией к осуществлению профессиональной	Общекультурные основы профессиональной деятельности, Социология, Модуль "Теоретические основы профессиональной деятельности", Психология, Педагогика, Математика, Физика, Геометрия, Теория вероятностей и математическая статистика, Математическая логика, Математический анализ и элементы теории функций, Алгебра, Элементарная математика, Информационные системы и сети, Информационные и коммуникационные технологии в образовании, Теория функций действительного переменного, Основы теории функций комплексного переменного, Архитектура профессионального компьютера и операционные системы, Профильное исследование в области математики, Профильное исследование в области математики, Элементарная алгебра, Элементы алгебры, Элементарная геометрия, Элементы	Текущий контроль успеваемости  Промежуточная аттестация	все	Выполнение контрольных работ Экзамен, зачёт

<p>деятельности.</p>	<p>геометрии, Информационные технологии в математике, Компьютерная алгебра, Дискретная математика, Избранные вопросы дискретной математики, Компьютерное моделирование, Моделирование информационных систем, Исследование операций, Методы оптимизации, Защита информации, Информационная безопасность, Организация исследовательской деятельности школьников, Intel - обучение для будущего, Дифференциальные уравнения, Дополнительные главы геометрии, История математики, История математического образования в России, Дифференциальная геометрия, Линии и поверхности в евклидовом пространстве, Числовые системы, Дополнительные главы алгебры, Основы искусственного интеллекта, Кибернетические системы деятельности человека, История информатики, История школьного курса информатики, Компьютерная графика, Трехмерная анимация, Открытые программные средства в школьном курсе информатики, Свободное программное обеспечение в обучении, Инновационные процессы в профильном образовании, Новые качество и методы обучения математике, Профессиональная деятельность учителя информатики, Теория и методика профильного обучения информатике, Классное руководство, Основы классного руководства, Методика обучения математике, Методика обучения информатике, Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, Производственная практика, Практика по получению профессиональных умений и опыта профессиональной деятельности, Преддипломная практика, Педагогическая практика интерна, Подготовка к сдаче и сдача государственного экзамена, Подготовка к защите и защита выпускной квалификационной работы</p>			
<p><b>ПК-11:</b> готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования.</p>	<p>Модуль "Научные основы учебно- профессиональной деятельности", Основы научной деятельности студента, Физика, Информатика, Элементарная математика, Математическая физика, Информационные системы и сети, Архитектура профессионального компьютера и операционные системы, Профильное исследование в области математики, Профильное исследование в области информатики, Элементарная алгебра, Элементы алгебры, Информационные технологии в математике, Компьютерная алгебра, Защита информации, Информационная безопасность, Основания геометрии, Дополнительные главы геометрии, Дифференциальная геометрия, Линии и поверхности в евклидовом пространстве, Основы искусственного интеллекта, Кибернетические системы деятельности человека, История информатики, История школьного курса информатики, Современные средства оценивания результатов обучения, Основы современной тестологии, Методика обучения математике. Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, Производственная практика, Практика по получению профессиональных умений и опыта профессиональной деятельности, Научно-исследовательская работа, Педагогическая практика интерна, Подготовка к сдаче и сдача государственного экзамена, Подготовка к защите и защита выпускной квалификационной работы</p>	<p>Текущий контроль успеваемости</p> <p>Промежуточная аттестация</p>	<p>все</p>	<p>Выполнение контрольных работ Экзамен, зачёт</p>



### 3. Фонд оценочных средств для промежуточной аттестации

3.1. Фонды оценочных средств включают: вопросы и задания к экзамену в 3 и 5 семестрах.

3.2. Оценочные средства

3.2.1. Оценочное средство 1 «Вопросы и задания к экзамену в 3 и 5 семестрах»

Критерии оценивания по оценочному средству «Вопросы и задания к экзамену в 3 и 5 семестрах»

Формируемые компетенции	Продвинутый уровень сформированности компетенций	Базовый уровень сформированности компетенций	Пороговый уровень сформированности компетенций
	(26 - 30 баллов) отлично	(22 - 25 баллов) хорошо	(15 - 21 балл)* Удовлетворительно
<b>ОК-6</b> - способность к самоорганизации и самообразованию.	Обучающийся свободно использует различные информационные ресурсы для поиска дополнительной информации и ответов на вопросы	Обучающийся фрагментарно использует различные информационные ресурсы для поиска дополнительной информации и ответов на вопросы	Обучающийся использует конкретные информационные ресурсы для поиска дополнительной информации и ответов на вопросы, но находит ответы с трудом
<b>ОПК-1</b> - готовность сознавать социальную значимость своей будущей профессии, обладать мотивацией к осуществлению профессиональной деятельности.	Обучающийся демонстрирует высокий уровень осознания значимости профессии и необходимости владения методами программирования в разных парадигмах	Обучающийся демонстрирует хороший уровень осознания значимости профессии и необходимости владения методами программирования в разных парадигмах	Обучающийся демонстрирует достаточный уровень осознания значимости профессии и необходимости владения методами программирования в разных парадигмах
<b>ПК-11:</b> готовность использовать систематизированные	Обучающийся демонстрирует высокий уровень готовности к использованию	Обучающийся демонстрирует хороший уровень готовности к использованию	Обучающийся демонстрирует достаточный уровень готовности к использованию



теоретические и практические знания для постановки и решения исследовательских задач в области образования.	полученных знаний для постановки и решения исследовательских задач по программированию	полученных знаний для постановки и решения исследовательских задач по программированию	полученных знаний для постановки и решения исследовательских задач по программированию
---	--	--	--

\*Менее 15 баллов – компетенция не сформирована

#### 4. Фонд оценочных средств для текущего контроля успеваемости

Фонды оценочных средств включает:

- 1) лабораторные работы по дисциплине
- 2) промежуточные тесты

4.2.1. Критерии оценивания по оценочным средствам 1-8. *Лабораторные работы №1 - 8 по модулям 1, 2*

Критерии оценивания	Количество баллов (вклад в рейтинг)
Выполнено до 50% заданий	0
Выполнено от 51% до 86% заданий	1,5
Выполнено от 87% до 100% заданий	3
Максимальный балл	3

4.2.2. Критерии оценивания по оценочным средствам 9-14. *Лабораторные работы № 9 - 14 по модулю 3*

Критерии оценивания	Количество баллов (вклад в рейтинг)
Выполнено до 50% заданий	0
Выполнено от 51% до 86% заданий	2
Выполнено от 87% до 100% заданий	4
Максимальный балл	4

4.2.3. Критерии оценивания по оценочным средствам 15 - 25. *Лабораторные работы № 15 - 25 по модулям 5, 6*

Критерии оценивания	Количество баллов (вклад в рейтинг)
Выполнено до 50% заданий	0
Выполнено от 51% до 86% заданий	3
Выполнено от 87% до 100% заданий	6

Максимальный балл	6
-------------------	---

4.2.4. Критерии оценивания по оценочным средствам 26 - 28. *Лабораторная работа № 26-28 “по модулю 6*

Критерии оценивания	Количество баллов (вклад в рейтинг)
Выполнено до 50% заданий	0
Выполнено от 51% до 86% заданий	4
Выполнено от 87% до 100% заданий	8
Максимальный балл	8

4.2.5. Критерии оценивания по оценочным средствам 29-30: *Лабораторная работа № 29-30 по модулю 7*

Критерии оценивания	Количество баллов (вклад в рейтинг)
Выполнено до 55% заданий	0
Выполнено от 56% до 86% заданий	6
Выполнено от 87% до 100% заданий	11
Максимальный балл	11



## 5.

# Оценочные средства (контрольно-измерительные материалы)

## 1. ВОПРОСЫ К ЭКЗАМЕНАМ

### Вопросы к экзамену в 3 семестре

1. Принципы структурного программирования. Основные алгоритмические конструкции.
2. Язык C/C++. Структура программы. Типы данных.
3. Язык C/C++. Библиотеки функций. Ввод-вывод данных. Примеры.
4. Язык C/C++. Организация ветвлений (if, :?). Конструкция множественного выбора (switch). Пример.
5. Язык C/C++. Организация циклов с постусловием и предусловием. Примеры.
6. Язык C/C++. Цикл с параметром. Его особенности и возможности. Примеры.
7. Язык C/C++. Указатели. Операторы работы с указателями. Арифметика указателей. Примеры.
8. Язык C/C++. Пользовательские функции. Способы передачи данных в функцию. Примеры.
9. Язык C/C++. Статические одномерные массивы. Создание массивов и обработка элементов массивов. Передача массивов в функцию. Примеры
10. Язык C/C++. Статические двумерные массивы. Создание массивов и обработка элементов массивов. Передача массивов в функцию. Примеры
11. Язык C/C++. Динамические одномерные массивы. Создание одномерных динамических массивов и обработка элементов массивов. Передача массивов в функцию. Связь массивов и указателей. Примеры
12. Язык C/C++. Динамические двумерные массивы. Создание и удаление двумерных динамических массивов и обработка элементов массивов. Передача массивов в функцию. Связь массивов и указателей. Примеры
13. Язык C/C++. Строки. Обработка строк. Связь строк, массивов и указателей. Пример.
14. Язык C/C++. Структуры. Описание структуры в программе. Массивы структур. Пример.
15. Язык C/C++. Динамические односвязные списки. Примеры процедур обработки списков.
16. Язык C/C++. Динамические списки. Стеки. Примеры процедур обработки стека.
17. Язык C/C++. Динамические списки. Очереди. Примеры процедур обработки очереди.
18. Язык C/C++. Динамические списки. Деки. Примеры процедур обработки двунаправленных списков.
19. Язык C/C++. Динамические списки. Бинарные деревья. Примеры процедур обработки деревьев.
20. Язык C/C++. Алгоритмы поиска и сортировки
21. Основы анализа эффективности алгоритмов. Трудоёмкость алгоритмов. Временная сложность алгоритмов. Классы сложности алгоритмов.
22. Основы анализа эффективности алгоритмов. Методы оценки ресурсной эффективности алгоритмов. Метод получения функции трудоёмкости. Пример.

### Задачи к экзамену в 3 семестре

#### Задача 1

Составить программу, проверяющую, является ли заданное натуральное число совершенным, то есть равным сумме своих положительных делителей, кроме самого этого числа.

#### Задача 2

В файле задана двоичная запись некоторого числа (не более 10ти знаков). Записать во второй файл соответствующее десятичное значение.

#### Задача 3

Одноклеточная амёба каждые 3 часа делится на 2 клетки. Определить, сколько амёб будет через 3, 6, 9, 12,..., 24 часа. Все результаты записать в файл.

#### Задача 4

Дано простое число. Составить функцию, которая будет находить следующее за ним простое число.

#### Задача 5

Даны две дроби  $A/B$  и  $C/D$  ( $A, B, C, D$  — натуральные числа). Составить программу для умножения дроби на дробь. Результат должен быть несократимой дробью.

#### Задача 6

В квадратной матрице вычислить сумму элементов, принадлежащих главной диагонали.

#### Задача 7

Проверить, имеется ли в заданном числовом массиве хотя бы одна пара чисел, совпадающих по величине. Если имеется напечатать их индексы, в противном случае сообщить об отсутствии совпадений.

#### Задача 8

Дана прямоугольная матрица. Получить новую матрицу путём деления всех элементов исходной матрицей на её максимальный по модулю элемент.

#### Задача 9

Дана строка. Группы символов, разделённые пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Вывести последовательность слов в обратном порядке.

*Задача 10*

В файле содержится прайс-лист автосалона. Информация о машине содержит: производителя, модель, стоимость, мощность. Вывести на печать список автомобилей, отсортированных по стоимости.

*Задача 11*

Сформировать список вещественных чисел. Вставить число 10 после каждого положительного элемента.

*Задача 12*

В текстовом файле записана без ошибок формула вида: цифра или S(формула, формула), или P(формула, формула), где  $S(a, b) = (a+b) \bmod 10$ ,  $P(a, b) = (a*b) \bmod 10$ . Вычислить значение данной формулы. Например:  $P(6, S(8, 4)) = 2$ .

*Задача 13*

Сформировать файл из вещественных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, кратные 5, а затем все остальные числа, сохраняя исходный порядок чисел в каждом случае.

*Задача 14*

Пусть L обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая определяет, есть ли в списке L хотя бы один элемент, который равен следующему за ним по кругу элементу.

*Задача 15*

Составить программу поиска количества ключей в бинарном дереве, значения которых больше заданного числа a.

### Вопросы к экзамену в 5 семестре

1. Функциональное программирование. Принципы функционального программирования. Язык Лисп и его диалекты.
2. Атомы и S-выражения. Списки и способы их записи. Функции присваивания и ввода-вывода данных. Примеры.
3. Язык Лисп. Математические функции. Числовые предикаты. Функции работы с данными. Примеры.
4. Язык Лисп. Основные конструкции: переменные, константы, функции, ветвления (IF, COND). Примеры
5. Язык Лисп. Рекурсивные функции. Пример. Механизм обработки программ, представленных S-выражениями.
6. Язык Лисп. Основные управляющие конструкции: оператор IF, конструкции WHEN и UNLESS. Конструкции AND, OR, NOT. Примеры.
7. Язык Лисп. Организация циклов: циклы DOLIST и DOTIMES, конструкция DO. Примеры.
8. Язык Лисп. Организация циклов: конструкция DO. Примеры.
9. Язык Лисп. Организация циклов. Цикл LOOP, ключевые слова цикла LOOP. Возврат значения из цикла LOOP. Примеры.
10. Язык Лисп. Функции. Именованные и безымянные функции. Способ определения и примеры.
11. Язык Лисп. Параметры функций: обязательные, необязательные, остаточные и именованные параметры
12. Пролог. Принципы логического программирования. Высказывания и предикаты. Работа программ на Прологе. Области применения языка.
13. Структура программы на Прологе. Назначение каждого раздела. Типы данных в Прологе. Примеры.
14. Предикаты, примеры описания предикатов. Пустые предикаты. Стандартные предикаты.
15. Поиск с возвратом. Его особенности. Пример. Пояснить на примере ход поиска с возвратом. Поиск всех альтернативных решений. Правила поиска с возвратом.
16. Отсечение и откат. Примеры.
17. Рекурсия. Случаи применения рекурсии. Откат с петлями (конструкция repeat. repeat :- repeat). Пример. Применение рекурсии для вычислений. Пример. Схема работы рекурсивной процедуры. Понятие стекового фрейма. Хвостовая рекурсия.
18. Списки. Два определения списка. Примеры определения списков. Обработка списков, этапы обработки. Основные задачи обработки списков. Примеры реализации перечисленных задач (по крайней мере трех).

### Задачи к экзамену в 5 семестре

*Задача 1 (Лисп).*

Посчитать значение математического выражения для допустимых значений x:

$$y = \frac{5}{9} \sqrt{\frac{\cos(x)}{\sin(x)}} - \frac{1}{x^3}$$

*Задача 2 (Лисп).*

Составить программу вычисления значения функции  $y = 12x^2 + 7x - 16$  при любом значении  $x$ .

*Задача 3( Лисп)*

Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа  $a, b, c, d$ , каждое из которых не превосходит восьми. На поле  $(a,b)$  расположен конь. Определить, угрожает ли он полю  $(c,d)$ .

*Задача 4( Лисп)*

Составить программу, которая в зависимости от порядкового номера дня сентября 2013 года выводит на экран название дня недели.

*Задача 5( Лисп)*

Известны даты рождения двух человек. Определить, кто из них старше. Для решения задачи написать именованную функцию

*Задача 6( Лисп)*

Определить в заданной строке количество знаков арифметических действий (+, -, и т.д.).

*Задача 7( Лисп)*

Прочитать из файла значения переменных  $x$  и  $y$ , а также список  $l$ . Написать рекурсивную функцию, заменяющую в списке  $l$  первое вхождение  $x$  на  $y$ . Список задать при вызове функции.

*Задача 8( Пролог)*

Известно, что Адам отец Каина и Авеля, Ева - мать Каина и Авеля. Отец является родителем, мать тоже является родителем. Если у двоих людей один родитель, то они братья. Написать Пролог-программу, которая делает вывод, что Каин и Авель – братья.

*Задача 9( Пролог)*

Люди живут в цветных домах. Те, кто живет в синих домах – спокойные, те, кто живет в красных – агрессивные, те, кто живет в зеленых – добрые. Петя и Ира живут в синем доме, Сара – добрая, Коля живет в красном доме. Написать программу, которая отвечает на вопросы. Кто является спокойным? Кто живет в красном доме? В зеленом?

*Задача 10( Пролог)*

Написать программу, которая осуществляет ввод с клавиатуры номера элемента и выводит значение  $x_n$  для последовательности  $x_n = 2x_{n-1} - 1, x_1 = 1$

*Задача 11( Пролог)*

Дано 2 списка. Инвертировать первый список, во втором удалить последний элемент. Результаты объединить.

*Задача 12( Пролог)*

В списке найти разность между максимальным и минимальным элементами.

## ЛАБОРАТОРНЫЕ РАБОТЫ

### Модуль 1. Базовые алгоритмические конструкции в языке C/C++

#### *Лабораторная работа 1.*

#### *Линейные программы. Функции ввода-вывода.*

**Задание 1.** Написать программы в соответствии со своим вариантом. Организовать ввод исходных данных с клавиатуры и вывод результата на экран.

**Задание 2.** Составить блок-схемы для записи алгоритма каждой задачи согласно варианту.

**Задание 3.** Подобрать набор тестовых значений *ключевых переменных* (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность программы. Тестовые значения должны содержать как стандартные, так и крайние значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы программы (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			
Шаг 2			
Шаг 3			

Для отчёта по лабораторной работе:

1. Работающую программу с комментариями продемонстрировать преподавателю.
2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», бакалавриат, задание 1: Лабораторная работа №1 Введение). Отчёт должен быть прикреплён в течение 2-х недель после лабораторной работы.

**Вариант 1.**

1. Найти площадь кольца по заданным внешнему и внутреннему радиусам.
2. Высота дерева  $k$  миллиметров. Определить, сколько полных метров и полных сантиметров составляет высота этого дерева.

**Вариант 2.**

1. Посчитать значение математического выражения для допустимых значений  $x$ :  $y = 7x^2 + \frac{2}{\sqrt{x^3}}$
2. Даны основание и высота равнобедренной трапеции. Найти её периметр.

**Вариант 3.**

1. Посчитать значение математического выражения для допустимых значений  $x$ :

$$y = \frac{5}{9} \sqrt{\frac{\cos(x)}{\sin(x)}} - \frac{1}{x^3}$$

2. Составить программу вычисления значения функции  $y = 7x^2 - 3x + 6$  при любом значении  $x$ .

**Вариант 4.**

1. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
2. Идет  $k$ -я секунда суток. Определить, сколько полных часов ( $h$ ) и полных минут ( $m$ ) прошло к этому моменту.

**Вариант 5.**

1. Составить программу вычисления значения функции  $y = \sqrt{\frac{2a + \sin|3a|}{3.56}}$  при допустимом значении  $a$ .
2. Треугольник задан координатами своих вершин. Найти периметр треугольника (считать, что вершины заданы верно, т.е. образуют треугольник).

**Вариант 6.**

1. Напишите программу, которая переводит сантиметры в дюймы.
2. Известно количество жителей в государстве и площадь его территории. Определить плотность населения в этом государстве.

**Вариант 7.**

1. Даны длины сторон прямоугольника. Найти его периметр и длину диагонали.
2. Масса тела  $k$  грамм. Определить, сколько полных тонн и полных килограмм составляет масса этого тела.

**Вариант 8.**

1. Составить программу вычисления значения функции  $y = \sin \frac{3.2 + \sqrt{1+x}}{|5x|}$  при допустимом значении  $x$ .
2. Написать программу, которая переводит температуру в градусах по Фаренгейту в градусы Цельсия по формуле:

$$C = \frac{5}{9} (F - 32)$$

где  $C$  – температура по Цельсию, а  $F$  – температура по Фаренгейту.

**Вариант 9.**

1. Составить программу вычисления значения функции  $y = 12x^2 + 7x - 16$  при любом значении  $x$ .
2. Даны стороны треугольника. Найти площадь треугольника.

**Вариант 10.**

1. Дан радиус окружности. Найти длину окружности и площадь круга.
2. Даны катеты прямоугольного треугольника. Найти гипотенузу.

**Лабораторная работа 2.  
Организация ветвлений.**

**Задание.** Написать программы, согласно варианту. При составлении алгоритма учесть правильность ввода исходных данных. Если данные введены неверно, то завершить программу с выводом соответствующего сообщения.

**Задание 2.** Составить блок-схемы для записи алгоритма каждой задачи согласно варианту.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность программы. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы программы (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			
Шаг 2			
Шаг 3			

Для отчёта по лабораторной работе:

1. Работающую программу с комментариями продемонстрировать преподавателю.
2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», бакалавриат, раздел 2: задания на лабораторные для 2 курса, задание 1: Лабораторная работа №2).

Отчёт должен быть прикреплён в течение 2-х недель после лабораторной работы.

**Вариант 1.**

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа a, b, c, d, каждое из которых не превосходит восьми. На поле (a,b) расположен ферзь. Определить, угрожает ли он полю (c,d).
2. Написать программу, которая будет проверять, попало ли случайно выбранное из отрезка [5;155] целое число в интервал (25;100) и сообщать результат на экран.

**Вариант 2.**

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа a, b, c, d, каждое из которых не превосходит восьми. На поле (a,b) расположен конь. Определить, угрожает ли он полю (c,d).
2. Дата некоторого дня определяется двумя натуральными числами: порядковым номером месяца и числом. Определить дату следующего дня.

**Вариант 3.**

1. Даны положительные числа a, b, c, x. Выясните, пройдет ли кирпич с ребрами a, b, c в квадратное отверстие со стороной x.
2. В старояпонском календаре был принят 60-летний цикл, состоящий из пяти 12-летних подциклов. Подциклы обозначались названиями цветов: зеленый, красный, желтый, белый и черный. Внутри каждого подцикла годы носили названия животных: крыса, бык, тигр, заяц, дракон, змея, лошадь, овца, обезьяна, курица, собака и свинья. Например 1984 год – год начала очередного цикла – назывался Годом Зеленой Крысы. Составить программу, которая по заданному номеру n года печатает его название по старояпонскому календарю при условии, что  $n \leq 1984$ .

**Вариант 4.**

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа a, b, c, d, e, f каждое из которых не превосходит восьми. На поле (a,b) расположен белый ферзь. На поле (c,d) расположен черный ферзь. Определить, может ли белый ферзь пойти на поле (e,f), не попав при этом под удар черного ферзя.
2. С начала 2000 года прошло n месяцев. Определить название наступившего месяца ( $n > 0$ ).

**Вариант 5.**

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа a, b, c, d, каждое из которых не превосходит восьми. На поле (a,b) расположен король. Определить, может ли он одним ходом попасть на поле (c,d).



2. В старояпонском календаре был принят 60-летний цикл, состоящий из пяти 12-летних подциклов. Подциклы обозначались названиями цветов: зеленый, красный, желтый, белый и черный. Внутри каждого подцикла годы носили названия животных: крыса, бык, тигр, заяц, дракон, змея, лошадь, овца, обезьяна, курица, собака и свинья. Например 1984 год – год начала очередного цикла – назывался Годом Зеленой Крысы. Составить программу, которая по заданному номеру  $n$  года печатает его название по старояпонскому календарю при условии, что  $n \geq 1984$ .

#### Вариант 6.

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа  $a, b, c, d$ , каждое из которых не превосходит восьми. На поле  $(a,b)$  расположена ладья. Определить, угрожает ли она полю  $(c,d)$ .
2. С начала 1980 года прошло  $n$  месяцев. Определить название наступившего месяца.

#### Вариант 7.

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа  $a, b, c, d$ , каждое из которых не превосходит восьми. На поле  $(a,b)$  расположена белая пешка. Определить, может ли она одним ходом попасть на поле  $(c,d)$ .
2. Составить программу, которая в зависимости от порядкового номера месяца выводит на экран количество дней в этом месяце (учесть високосный год или нет, проверить правильность ввода данных, если номер месяца введён неверно, завершить программу).

#### Вариант 8.

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа  $a, b, c, d$ , каждое из которых не превосходит восьми. На поле  $(a,b)$  расположен слон. Определить, угрожает ли он полю  $(c,d)$ .
2. Составить программу, которая в зависимости от порядкового номера месяца выводит на экран его название.

#### Вариант 9.

1. Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми: первое число - номер вертикали (при счете слева направо), второе - номер горизонтали (при счете снизу вверх). Даны натуральные числа  $a, b, c, d, e, f$  каждое из которых не превосходит восьми. На поле  $(a,b)$  расположена белая ладья. На поле  $(c,d)$  расположен черный слон. Определить, может ли ладья пойти на поле  $(e,f)$ , не попав при этом под удар слона.
2. Составить программу, которая в зависимости от порядкового номера дня сентября 2013 года выводит на экран название дня недели.

#### Вариант 10.

1. Составьте программу, проверяющую, верно ли утверждение, что введённое вами число является чётным.
2. Мужчина заполняет в военкомате анкету и программа должна в зависимости от указанного им возраста выводить разные подсказки, а именно:
  - Если указан возраст от 18 и до 27 лет, то сообщать, что заполняющий подлежит призыву на срочную службу или может служить по контракту.
  - Если указан возраст от 28 до 59 лет, то сообщать, что заполняющий может служить по контракту.
  - Если указан возраст менее 18 или более 59 лет, то сообщать о том, что заполняющий находится в непризывном возрасте.

Если указан неположительный возраст или возраст более 120 лет, то сообщить об ошибке.

#### Лабораторная работа №3.

##### Организация циклов в языке C.

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке C, составление программы с использованием сравнений, ветвлений, организацией циклов.

**Задание 1.** Написать программы, согласно варианту. Указание: для каждой задачи предусмотреть проверку правильности ввода данных (если данные вводятся с клавиатуры). Для получения дополнительных баллов – организовать в программе текстовое меню с пунктами: ввод данных, расчёт, вывод результата, выход.

**Задание 2.** Составить блок-схемы для записи алгоритма каждой задачи согласно варианту.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность программы. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы программы (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			

<b>Шаг 2</b>			
<b>Шаг 3</b>			

Для отчёта по лабораторной работе:

3. Работающую программу с комментариями продемонстрировать преподавателю.
4. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования»: Лабораторная работа №3).

Отчёт должен быть прикреплён в течение 2-х недель после лабораторной работы.

#### **Вариант 1.**

1. Дано натуральное число. Определить количество его цифр, кратных  $z$  (значение  $z$  вводится с клавиатуры).
2. Дано натуральное число  $n \geq 2$ . Составить программу разложения этого числа на простые множители. Каждый простой множитель должен быть выведен столько раз, сколько он встречается в разложении. Например: для  $n=124$  список простых множителей: 2, 2, 31.

#### **Вариант 2.**

1. Исходное данное – натуральное число  $q$ , выражающее площадь. Написать программу для нахождения всех таких прямоугольников, площадь которых равна  $q$  и стороны выражены натуральными числами.
2. Даны натуральные числа  $N$  и  $p$ . Получить все натуральные числа, меньшие  $N$  и взаимно простые с  $p$ .

#### **Вариант 3.**

1. Составить программу, проверяющую, является ли последовательность из  $n$  чисел, вводимых с клавиатуры, возрастающей.
2. Даны натуральное число  $n$  и целые числа  $x_1, x_2, \dots, x_n$ . Найти номер минимального из чисел (массив не использовать).

#### **Вариант 4.**

1. Найти сумму цифр числа.
2. Найти произведение элементов вектора (массивы не использовать, значения перемножать по мере ввода).

#### **Вариант 5.**

1. Даны натуральные числа  $m$  и  $n$ . Получить все меньшие  $n$  натуральные числа, квадрат суммы цифр которых равен  $m$ .
2. В данном натуральном числе переставить цифры таким образом, чтобы получилось наименьшее число записанное этими же цифрами.

#### **Вариант 6.**

1. Составить программу, проверяющую, является ли заданное натуральное число совершенным, то есть равным сумме своих положительных делителей, кроме самого этого числа.
2. Составить программу, выясняющую, является ли данное число палиндромом, то есть таким, десятичная запись которого читается одинаково слева направо и справа налево.

#### **Вариант 7.**

1. Найти самую большую цифру целого числа.
2. Даны натуральное число  $n$  и вещественные числа  $x_1, x_2, \dots, x_n$ . Найти минимальное из вещественных чисел (массив не использовать).

#### **Вариант 8.**

1. Найти сумму элементов вектора (массивы не использовать, значения складывать по мере ввода).
2. Найти все трехзначные числа, удовлетворяющие условию: все цифры различны.

#### **Вариант 9.**

1. Для натурального числа определить какая из его цифр больше: первая или последняя.
2. Напечатать числа в виде следующей таблицы:

```
5 5 5 5 5
6 6 6 6
7 7 7
8 8
9
```

#### **Вариант 10.**

1. Дано натуральное число. Получить все делители этого числа.
2. Составить программу, выясняющую, является ли данное число палиндромом, то есть таким, десятичная запись которого читается одинаково слева направо и справа налево.

#### **Лабораторная работа №4.**

#### **Работа с файлами в языке C.**

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке C, составление программы с использованием ветвлений, организацией циклов, с чтением начальных данных из файла и записью результата в файл.

**Задание 1.** Написать программы, согласно варианту.

Указание: при решении задач ввод данных в обязательном порядке должен производиться из файла и вывод результата также должен быть выполнен в файл.

**Задание 2.** Составить блок-схемы для записи алгоритма каждой задачи согласно варианту.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность программы. Тестовые значения должны содержать как стандартные, так и крайние значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы программы (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			
Шаг 2			
Шаг 3			

Для отчёта по лабораторной работе:

1. Работавшую программу с комментариями продемонстрировать преподавателю.
  2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», задание 4: Лабораторная работа №4).
- Отчёт должен быть прикреплён в течение 2-х недель после лабораторной работы.

**Вариант 1.**

1. В файле записана последовательность целых чисел. Найти наименьшее из чисел и записать результат во второй файл.
2. Вычислить количество точек с целочисленными координатами, находящихся в круге радиуса  $R$  ( $R > 0$ ). Радиус круга прочитать из файла с входными данными. Количество точек и их координаты записать в выходной файл.

**Вариант 2.**

1. В файле записана последовательность целых чисел. Записать во второй файл только нечётные числа.
2. Напечатать в файл в возрастающем порядке все трёхзначные числа, в десятичной записи которых нет одинаковых цифр.

**Вариант 3.**

1. В файле записана последовательность целых чисел. Записать во второй файл только числа, кратные 4.
2. Создать файл с таблицей ASCII кодов строчных букв латинского алфавита.

**Вариант 4.**

1. В файле записана последовательность целых чисел. Записать во второй файл квадраты чисел.
2. Записать в файл ряд чисел Фибоначчи. Количество чисел прочитать из файла с исходными данными.

**Вариант 5.**

1. В файле записана последовательность целых чисел. Записать во второй файл только числа, кратные 7.
2. В файле задана двоичная запись некоторого числа (не более 10ти знаков). Записать во второй файл соответствующее десятичное значение.

**Вариант 6.**

1. В файле записана последовательность целых чисел. Найти наибольшее из чисел и записать результат во второй файл.
2. Составить программу для вычисления значений функции  $F(x)=x-\sin(x)$  на отрезке  $[a, b]$  с шагом  $h$ . Значения  $a$ ,  $b$  и  $h$  прочитать из файла с входными данными. Результат представить в выходном файле в виде таблицы, первый столбец которой — значения аргумента, второй — соответствующие значения функции.

**Вариант 7.**

1. В файле записана последовательность целых чисел. Записать во второй файл сумму последовательности.
2. У гусей и кроликов вместе 64 лапы. Сколько может быть кроликов и гусей (указать все сочетания)? Все результаты записать в файл.

**Вариант 8.**

1. В файле записана последовательность целых чисел. Найти среднее арифметическое чисел и записать результат во второй файл.
2. Одноклеточная амёба каждые 3 часа делится на 2 клетки. Определить, сколько амёб будет через 3, 6, 9, 12, ..., 24 часа. Все результаты записать в файл.

**Вариант 9.**

1. В файле записана последовательность целых чисел. Записать во второй файл их удвоенные значения.
2. Вычислить сумму кодов всех символов, которые в цикле читаются из файла. Результат записать в выходной файл.

**Вариант 10.**

1. В файле записана последовательность целых чисел. Записать во второй файл только чётные числа.
2. Составить программу, которая формирует файл с таблицей умножения и сложения натуральных чисел в десятичной системе счисления.

**Модуль 2. Указатели и массивы в С.**  
**Лабораторная работа №5.**  
**Указатели.**

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке С, составление программы с использованием указателей.

**Часть 1. Общая.**

**Задание 1.**

Дан фрагмент программы:

```
int main ()
{
int *Q;
int j;
Q=&j;
...}
```

Дополнить программу следующими командами:

- а) проинициализировать переменную j произвольным значением и вывести это значение через указатель Q.
- б) вывести на экран адрес переменной j через саму переменную и через указатель на неё (т.е. два раза)
- в) поместить по адресу памяти, хранимому в Q новое значение и вывести на экран значение переменной j;

**Задание 2.**

Написать программу по следующему алгоритму:

- а) описать два указателя: целочисленный и вещественный (double или float)
- б) для целочисленного указателя выделить память с помощью оператора new, вывести полученный адрес на экран, поместить по этому адресу значение (запросив его у пользователя), и выполнить над значением следующие действия: значение\*2-120+значение/3, вывести результат и освободить память используя delete.
- в) для вещественного указателя выделить память с помощью оператора new, вывести полученный адрес на экран, поместить по этому адресу значение (прочитав его из файла), и выполнить над значением следующие действия: sin(знач)/2+cos(знач)/3, вывести результат и освободить память используя оператор delete.

**Часть 2. По вариантам.**

**Задание 1.** Написать программу, согласно варианту.

Указание: при решении задач одна из переменных должна быть указателем. Если используется указатель, например, для хранения очередного прочитанного значения, то статическую переменную для этих целей не применять. Т.е. статические и динамические переменные не должны дублировать друг друга.

**Задание 2.** Составить блок-схему для записи алгоритма задачи согласно варианту.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность программы. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы программы (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
<b>Шаг 1</b>			
<b>Шаг 2</b>			
<b>Шаг 3</b>			

Для отчёта по лабораторной работе:

5. Работавшую программу с комментариями продемонстрировать преподавателю.
6. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», бакалавриат, задание 5: Лабораторная работа №5).

Отчёт должен быть прикреплён в течение 2-х недель после лабораторной работы.

Для первой (общей) части лабораторной письменный отчет не требуется.

**Вариант 1.**

1. В файле записана последовательность целых чисел. Найти среднее арифметическое чисел и записать результат во второй файл. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

**Вариант 2.**

1. В файле записана последовательность целых чисел. Записать во второй файл только нечётные числа. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

**Вариант 3.**

1. В файле записана последовательность целых чисел. Записать во второй файл только числа, кратные 4. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

#### Вариант 4.

1. Одноклеточная амёба каждые 3 часа делится на 2 клетки. Определить, сколько амёб будет через 3, 6, 9, 12,..., 24 часа. Все результаты записать в файл. Указание: переменную, предназначенную для расчёта количества амёб, объявить как указатель.

#### Вариант 5.

1. В файле записана последовательность целых чисел. Записать во второй файл квадраты чисел. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

#### Вариант 6

1. В файле записана последовательность целых чисел. Найти наименьшее из чисел и записать результат во второй файл. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

#### Вариант 7

1. В файле записана последовательность целых чисел. Записать во второй файл только числа, кратные 7. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

#### Вариант 8

1. В файле записана последовательность целых чисел. Записать во второй файл только чётные числа. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

#### Вариант 9

1. В файле записана последовательность целых чисел. Записать во второй файл их удвоенные значения. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

#### Вариант 10

1. В файле записана последовательность целых чисел. Найти наибольшее из чисел и записать результат во второй файл. Указание: переменную, предназначенную для временного хранения чисел в программе, объявить как указатель.

### Лабораторная работа №6.

#### Функции.

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке C, составление программы с использованием вспомогательных функций. Обязательное условие: использование вспомогательной функции при решении задачи.

#### Общая часть.

В задаче вычисления квадратного корня методом Ньютона-Рафсона, рассмотренной на лекции, видоизменить код так, чтобы:

1. Параметры в функции передавались по указателю.
2. Параметры в функции передавались по ссылке.

#### По вариантам.

**Задание 1.** Написать программу, согласно варианту.

**Задание 2.** Составить блок-схему для записи алгоритма работы функции.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность функции. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы функции (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			
Шаг 2			
Шаг 3			

Для отчёта по лабораторной работе:

1. Работавшую программу с комментариями продемонстрировать преподавателю.
2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования»: лабораторная работа №6).

Отчёт должен быть прикреплен в течение 2-х недель после лабораторной работы.

Для первой (общей) части лабораторной письменный отчет не требуется.

#### Вариант 1.

1. Даны две дроби  $A/B$  и  $C/D$  ( $A, B, C, D$  — натуральные числа). Составить программу для деления дроби на дробь. Результат должен быть несократимой дробью.
2. Составить функцию для нахождения наименьшего нечетного натурального делителя  $k$  ( $k \neq 1$ ) любого заданного натурального числа  $n$ .

#### Вариант 2.

1. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше.

2. Дано простое число. Составить функцию, которая будет находить следующее за ним простое число.

#### Вариант 3.

1. Даны две дроби  $A/B$  и  $C/D$  ( $A, B, C, D$  — натуральные числа). Составить программу для сложения дробей. Результат должен быть несократимой дробью.

2. Имеется часть катушки с автобусными билетами. Номер билета шестизначный. Составить программу, определяющую количество счастливых билетов на катушке, если меньший номер билета —  $N$ , больший —  $M$ .

#### Вариант 4.

1. Даны натуральные числа  $a, b$ , обозначающие соответственно числитель и знаменатель дроби. Сократить дробь, т.е. получить правильную несократимую дробь.

2. Составить программу для нахождения наибольшего общего делителя и наименьшего общего кратного двух натуральных чисел

$$(\text{НОК}(A, B) = \frac{A \cdot B}{\text{НОД}(A, B)})$$

#### Вариант 5.

1. Два простых числа называются «близнецами», если отличаются друг от друга на 2 (например: 41 и 43). Напечатать все пары чисел-«близнецов», не превышающих значение 200.

2. Составить программу для нахождения наибольшего общего делителя четырех натуральных чисел.

#### Вариант 6.

1. Три квадратных уравнения заданы своими коэффициентами. Определить, сколько из них имеют вещественные корни.

2. Составить программу для нахождения наименьшего общего кратного трех натуральных чисел.

#### Вариант 7.

1. Получить все 6-значные счастливые числа, т.е. те, у которых сумма первых трех цифр равна сумме последних трех.

2. Составить программу, проверяющую, являются ли данные три числа взаимно простыми.

#### Вариант 8.

1. Даны две дроби  $A/B$  и  $C/D$  ( $A, B, C, D$  — натуральные числа). Составить программу для умножения дроби на дробь. Результат должен быть несократимой дробью.

2. Написать программу для вычисления суммы факториалов всех нечетных чисел от 1 до 9.

#### Вариант 9.

1. Даны стороны двух треугольников. Найти сумму их площадей и периметров.

2. Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров.

#### Вариант 10.

1. Составить программу, определяющую, в каком из данных двух чисел больше цифр.

2. Составить программу для вычисления суммы факториалов всех четных чисел от  $a$  до  $b$

### Лабораторная работа №7.

#### Статические и динамические массивы.

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке C, составление программ обработки статических и динамических одномерных и двумерных массивов, программ с передачей указателей в функции.

#### Указания:

1) при решении задач с одномерными массивами **обязательно** использование вспомогательных функций с передачей параметров по указателю;

2) из двух задач в вариантах одна должна быть решена с помощью статических массивов, а вторая — с помощью динамических массивов.

#### Общая часть.

1. Решить задачу умножения матриц, рассмотренную на лекции 7 с использованием динамических двумерных массивов.

#### По вариантам.

**Задание 1.** Написать программу, согласно варианту.

**Задание 2.** Составить блок-схему для записи алгоритма работы **функции** (для двумерных массивов блок-схему составить для цикла, выполняющего основную вычислительную задачу), реализующей часть алгоритма.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность **функции**. Для двумерных массивов набор тестовых значений подбирается для цикла, выполняющего основную вычислительную задачу. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы **функции** (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			

Шаг 2			
Шаг 3			

Для отчёта по лабораторной работе:

1. Работающую программу с комментариями скопировать в элемент «Домашнее задание» темы «Статические и динамические массивы».
2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», задание 7: Лабораторная работа №7).

**Вариант 1.**

1. Дан массив чисел  $a_1, a_2, \dots, a_n$ . Получить массив  $b_1, b_2, \dots, b_n$ , в котором  $b_i = a_1 + a_2 + \dots + a_i$ .
2. Определить номера строк прямоугольной матрицы, содержащие только положительные элементы.

**Вариант 2.**

1. Многочлен степени  $n$  задан массивом своих коэффициентов. Подсчитать коэффициенты производной многочлена.
2. В квадратной матрице вычислить сумму элементов, принадлежащих главной диагонали.

**Вариант 3.**

1. Задан числовой массив  $a_1, a_2, \dots, a_n$ . Вычислить сумму произведений всех пар соседних чисел.
2. Определить номера строк прямоугольной матрицы, содержащие только отрицательные элементы.

**Вариант 4.**

1. Определить в числовом массиве  $a_1, a_2, \dots, a_n$  число соседств из двух чисел разного знака.
2. В квадратной матрице вычислить сумму элементов, принадлежащих побочной диагонали.

**Вариант 5.**

1. Проверить, имеется ли в заданном числовом массиве хотя бы одна пара чисел, совпадающих по величине. Если имеется напечатать их индексы, в противном случае сообщить об отсутствии совпадений.
2. Напечатать номера строк и столбца прямоугольной матрицы, на пересечении которых стоит максимальный элемент.

**Вариант 6.**

1. Вычислить скалярное произведение двух векторов.
2. В прямоугольной матрице найти сумму элементов, принадлежащих первой строке, последней строке, первому столбцу и последнему столбцу.

**Вариант 7.**

1. Для одномерного числового массива вычислить разность между наибольшим и наименьшим элементами.
2. Напечатать номера строк и столбца прямоугольной матрицы, на пересечении которых стоит минимальный элемент.

**Вариант 8.**

1. В числовом массиве подсчитать сумму его элементов, расположенных между максимальным и минимальным элементами.
2. В прямоугольной матрице  $A(m,n)$  заменить нечётные строки вектором  $B(n)$ .

**Вариант 9.**

1. В заданном числовом массиве вычислить сумму произведений всех троек соседних чисел.
2. Дана прямоугольная матрица. Получить новую матрицу путём деления всех элементов исходной матрицей на её максимальный по модулю элемент.

**Вариант 10.**

1. Дан числовой массив. Определить, сколько раз в нем встречается максимальный по величине элемент.
2. В прямоугольной матрице  $A(m,n)$  заменить четные столбцы вектором  $B(m)$ .

**Лабораторная работа №8.**

**Строки.**

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке C, составление программ обработки строк с использованием указателей, программ с передачей указателей в функции.

**Задание 1.** Написать программу, согласно варианту.

**Задание 2.** Составить блок-схему для записи алгоритма работы **функции**, реализующей часть алгоритма.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность **функции**. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы **функции** (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			

Шаг 2			
Шаг 3			

Для отчёта по лабораторной работе:

1. Работающую программу с комментариями продемонстрировать преподавателю.
2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования» С Лабораторная работа №8).

**Вариант 1.**

1. Написать программу выделения из строки заданного количества слов, начиная со слова с заданным номером.
2. Дано слово a. Получить слово b, образованное нечётными буквами слова a.

**Вариант 2.**

1. Дана строка. Заменить в ней все строчные (латинские и русские) буквы соответствующими заглавными. Программа должна запросить с клавиатуры строку и, при необходимости, дополнительные данные (подсказка: ASCII-коды для заглавных букв отстают от значений соответствующих строчных букв на 32 позиции.).
2. Напечатать в алфавитном порядке все согласные буквы, входящие в текст.

**Вариант 3.**

1. Дана строка. Группы символов, разделённые пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Вывести последовательность слов в обратном порядке.
2. В предложении заменить каждую точку многоточием.

**Вариант 4.**

1. Дана строка символов, состоящая из русских слов, разделённых пробелами. В каждом чётном по порядку слове удалить все нечётные буквы, а в каждом нечётном слове удалить все чётные буквы. Полученные слова вывести на экран.
2. Дано предложение. Заменить в нем букву a на букву z.

**Вариант 5.**

1. Написать программу разбиения строки на две части: до первого вхождения заданного символа и после него.
2. Найти все слова, которые встречаются хотя бы в одном из двух заданных предложений.

**Вариант 6.**

1. Написать программу удаления из строки заданного количества символов, начиная с заданной позиции.
2. Дан текст. Напечатать все имеющиеся в нем цифры.

**Вариант 7.**

1. Написать программу определения в строке номера позиции слова с заданным номером.
2. Найти все слова, которые встречаются в каждом из двух заданных предложений.

**Вариант 8.**

1. Написать программу поиска последнего вхождения в строку заданной подстроки.
2. Дан текст. Определить, каких букв в нем больше k или m.

**Вариант 9.**

1. Написать программу, которая доводит длину строки до заданной, вставляя пробелы между словами.
2. Заданы текст и три ключевых слова. Определить, встречаются или нет ключевые слова в тексте.

**Вариант 10.**

1. Написать программу удаления из строки слова с заданным номером.
2. Дана последовательность символов, в начале которой имеется некоторое количество одинаковых символов. Определить это количество.

### Модуль 3. Структуры и списки в C.

#### Лабораторная работа №10.

##### Структуры.

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке C\C++, составление программ обработки массивов структур.

**Указание.** Исходные данные поместить в файл output.txt. Результат – в файл input.txt.

**Задание 1.** Написать программу, согласно варианту.

**Задание 2.** Составить блок-схему для записи алгоритма работы **функции**, реализующей часть алгоритма.

**Задание 3.** Подобрать набор тестовых значений ключевых переменных (не менее пяти для каждой задачи), с помощью которых вы можете проверить работоспособность **функции**. Тестовые значения должны содержать как стандартные, так и краевые значения.

**Задание 4.** Для одного из тестовых значений составить таблицу изменения значений ключевых переменных на каждом шаге работы **функции** (трассировку).

*Форма таблицы трассировки*

	Переменная 1	Переменная 2	Переменная 3
Шаг 1			
Шаг 2			



<b>Шаг 3</b>			
--------------	--	--	--

Для отчёта по лабораторной работе:

1. Работающую программу с комментариями продемонстрировать преподавателю.
2. Задания 2-4 оформить в электронном виде в формате doc или docx и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», лабораторная работа 9: структуры).

**Вариант 1.**

1. Используя массив структур создать список студентов группы и выяснить, сколько человек сдали сессию без троек. В структуре должно быть не менее 3-х полей.
2. Имеются сведения о кубиках: размер каждого кубика (длина ребра в см), его цвет (красный, желтый, зеленый или синий) и материал (дерево, металл, картон). Найти количество деревянных кубиков с ребром 3 см. Использовать массив структур.

**Вариант 2.**

1. Используя массив структур создать список студентов группы и выяснить, сколько человек живёт в общежитии и вне его. В структуре должно быть не менее 3-х полей.
2. В файле содержится расписание авиаперелётов. Информация об авиаперелёте содержит: номер рейса, дату и время вылета и пункт назначения. Вывести на печать рейсы, вылетающие в заданный пункт назначения.

**Вариант 3.**

1. Используя массив структур создать список студентов группы и вывести список неуспевающих. В структуре должно быть не менее 3-х полей.
2. Имеются сведения о кубиках: размер каждого кубика (длина ребра в см), его цвет (красный, желтый, зеленый или синий) и материал (дерево, металл, картон). Найти количество зеленых картонных кубиков. Использовать массив структур.

**Вариант 4.**

1. Используя массив структур создать расписание движения поездов на сутки и вывести те из них, которые отправляются после 6 часов вечера. В структуре должно быть не менее 3-х полей.
2. В файле содержится прайс-лист автосалона. Информация о машине содержит: производителя, модель, стоимость, мощность. Вывести на печать список автомобилей, отсортированных по стоимости.

**Вариант 5.**

1. Используя массив структур создать расписание движения поездов на сутки и вывести те из них, которые отправляются в западном направлении. В структуре должно быть не менее 3-х полей.
2. В файле содержится расписание авиаперелётов. Информация об авиаперелёте содержит: номер рейса, дату и время вылета и пункт назначения. Вывести на печать список рейсов, отсортированных по дате вылета.

**Вариант 6.**

1. Используя массив структур создать список студентов группы и вывести отдельно список юношей и список девушек. В структуре должно быть не менее 3-х полей.
2. В файле содержится прайс-лист автосалона. Информация о машине содержит: производителя, модель, стоимость, мощность. Вывести на печать список автомобилей, заданного производителя.

**Вариант 7.**

1. Используя массив структур создать список студентов группы и вывести список студентов, имеющих только тройки за последнюю сессию. В структуре должно быть не менее 3-х полей.
2. В файле содержится прайс-лист автосалона. Информация о машине содержит: производителя, модель, стоимость, мощность. Вывести на печать список автомобилей, отсортированных по мощности.

**Вариант 8.**

1. Используя массив структур создать список книг домашней библиотеки и отсортировать его по алфавиту относительно имени автора. В структуре должно быть не менее 3-х полей.
2. В файле содержится расписание поездов. Информация о поезде содержит: номер поезда, дату и время отправления, пункт назначения, количество свободных мест. Вывести на печать список поездов, отсортированных в порядке убывания по количеству свободных мест.

**Вариант 9.**

1. Используя массив структур создать список книг домашней библиотеки и напечатать названия и авторов детективов. В структуре должно быть не менее 3-х полей.
2. В файле содержится расписание авиаперелётов. Информация об авиаперелёте содержит: номер рейса, дату и время вылета и пункт назначения. Вывести на печать список рейсов, отсортированных по пункту назначения.

**Вариант 10.**

1. Используя массив структур создать список жильцов некоторого дома и напечатать фамилии и номер квартиры тех, кто имеет долг за жилье. В структуре должно быть не менее 3-х полей.
2. В файле содержится расписание авиаперелётов. Информация об авиаперелёте содержит: номер рейса, дату и время вылета и пункт назначения. Вывести на печать рейсы, вылетающие позже 14-00.

**Лабораторная работа №10.  
Односвязные списки.**

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке С, составление программ обработки односвязных списков.

**Указание.** Исходные данные поместить в файл input.txt. Результат – в файл output.txt.

**Задание 1.** Написать программу, согласно варианту.

**Задание 2.**

Составить серию схем (см. материалы лекций), отражающих основные действия над списком в задаче (например, вставка элемента в определенное задачей место списка, удаление элемента, просмотр списка, перестановка элементов и т.д.).

Схемы должны в точности соответствовать реальной программе.

Порядок соединений и разрывов связей между элементами списка должен быть пронумерован и прокомментирован.

Для отчёта по лабораторной работе:

1. Работавшую программу с комментариями продемонстрировать преподавателю.
2. Листинг программы включить в отчет.
3. Задания 2-4 оформить в электронном виде в формате doc, docx или pdf, файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», ЯиМП, Лабораторная работа №10).

**Вариант 1.**

1. Сформировать список из вещественных чисел. Упорядочить список по убыванию.
2. Сформировать список из символов, оканчивающийся точкой. Вывести его на экран в обратном порядке.

**Вариант 2.**

1. Сформировать список из натуральных чисел. Заменить минимальные элементы списка числом 1.
2. Сформировать список вещественных чисел. Переставить местами первый и последний элементы списка, вывести результат на экран.

**Вариант 3.**

1. Сформировать список вещественных чисел. Найти среднее арифметическое элементов списка.
2. Сформировать список из натуральных чисел. Заменить максимальные элементы списка числом 100.

**Вариант 4.**

1. Сформировать список вещественных чисел. Найти количество чисел, больших 100.
2. Сформировать список из натуральных чисел. Найти среднее арифметическое элементов списка и выяснить, есть ли в списке такой элемент.

**Вариант 5.**

1. Сформировать список вещественных чисел. Вставить число 10 после каждого положительного элемента.
2. Сформировать список из целых чисел. После каждого отрицательного элемента вставить число -3.

**Вариант 6.**

1. Сформировать два списка L1 и L2. Скопировать в список L1 все нечётные элементы списка L2 вслед за каждым вхождением заданного элемента.
2. Сформировать список символов, оканчивающийся точкой. Заменить в списке символ “А” на символ “0”.

**Вариант 7.**

1. Сформировать список символов, оканчивающийся точкой. Заменить в списке символ “F” на символ “4”.
2. Сформировать два списка L1 и L2. Скопировать в список L1 все элементы списка L2 вслед за каждым вхождением заданного элемента.

**Вариант 8.**

1. Сформировать два списка L1 и L2. Скопировать в список L1 все чётные элементы списка L2 вслед за каждым вхождением заданного элемента.
2. Сформировать список из натуральных чисел. Проверить наличие в списке заданного элемента.

**Вариант 9.**

1. Сформировать список из натуральных чисел. Найти произведение элементов списка.
2. Сформировать список из вещественных чисел. Упорядочить список по возрастанию

**Вариант 10.**

1. Сформировать список из натуральных чисел. Удалить из списка отрицательные элементы.
2. Сформировать список из натуральных чисел. Удалить из списка все элементы, большие 50-ти.

**Лабораторная работа №11.**

**Язык C/C++. Стеки. Очереди**

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы со стеками и очередями.

**Указания.** Для решения первой задачи использовать стек, для решения второй - очередь. Исходные данные поместить в файл input.txt. Результат – в файл output.txt.

**Задание 1.**

Написать программу, согласно варианту.

**Задание 2.**

Составить серию схем (см. материалы лекций), отражающих основные действия над списком в вашей задаче (например, вставка элемента в определенное место списка, удаление элемента, просмотр списка, перестановка элементов и т.д.).

Схемы должны в точности соответствовать реальной программе.

Порядок соединений и разрывов связей между элементами списка должен быть пронумерован и прокомментирован.

Для отчёта по лабораторной работе:

1. Работающую программу с комментариями продемонстрировать преподавателю.
2. Листинг программы включить в отчет.

Задания оформить в электронном виде в формате doc, docx или pdf и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», Лабораторная работа №11).

#### **Вариант 1**

1. Используя стек напечатать содержимое текстового файла, выписывая символы каждой его строки в обратном порядке.
2. Сформировать файл из вещественных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа до отрезка  $[a; b]$ , а затем все числа из отрезка  $[a; b]$ , сохраняя исходный порядок чисел в каждом случае.

#### **Вариант 2**

1. В текстовом файле записана без ошибок формула вида: цифра или  $M(\text{формула}, \text{формула})$ , или  $m(\text{формула}, \text{формула})$ , где  $M$  обозначает функцию  $\max$ ,  $m$  -  $\min$ . Вычислить значение данной формулы. Например:  $(5, m(6, 8)) = 6$
2. Сформировать файл из символов и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все символы, отличные от цифр, а затем все цифры, сохраняя исходный порядок в каждом случае.

#### **Вариант 3**

1. В текстовом файле записана без ошибок формула вида: цифра или  $R(\text{формула}, \text{формула})$ , или  $L(\text{формула}, \text{формула})$ , где  $R$  обозначает функцию взять правое число,  $L$  – левое число. Вычислить значение данной формулы. Например:  $R(8, R(3, L(4,5))) = 4$ .
2. Сформировать файл из натуральных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все однозначные числа, а затем все двузначные числа, сохраняя исходный порядок в каждом случае.

#### **Вариант 4**

1. В текстовом файле записан текст, сбалансированный по круглым скобкам. Написать программу, которая для каждой открывающей и, соответственно, закрывающей пары скобок печатает номера их позиций в тексте, упорядочив пары в порядке возрастания номеров позиций закрывающих скобок. Например: для текста  $a+(45-f(x)*(b-c))$  надо напечатать 8, 10; 12, 16; 3, 17.
2. Сформировать файл из символов и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все символы, отличные от знаков препинания и цифр, затем все знаки препинания, а затем все цифры, сохраняя исходный порядок в каждом случае.

#### **Вариант 5**

1. В текстовом файле записана без ошибок формула вида: цифра или  $S(\text{формула}, \text{формула})$ , или  $P(\text{формула}, \text{формула})$ , где  $S(a, b) = (a+b) \bmod 10$ ,  $P(a, b) = (a*b) \bmod 10$ . Вычислить значение данной формулы. Например:  $P(6, S(8, 4)) = 2$ .
2. Сформировать файл из натуральных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, большие  $b$ , затем все числа, меньшие  $a$ , а затем все числа из отрезка  $[a; b]$ , сохраняя исходный порядок в каждом случае.

#### **Вариант 6**

1. В текстовом файле записана без ошибок формула вида: цифра или  $m(\text{формула}, \text{формула})$ , или  $p(\text{формула}, \text{формула})$ , где  $m(a, b) = (a+b) \bmod 10$ ,  $p(a, b) = (a+b) \div 10$ . Вычислить значение данной формулы. Например:  $m(9, p(3, 5)) = 9$ .
2. Сформировать файл из натуральных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, кратные 5, затем все нечетные числа, не кратные 5, сохраняя исходный порядок в каждом случае.

#### **Вариант 7**

1. Сформировать файл из натуральных чисел и за один просмотр файла напечатать элементы файла в следующем порядке: сначала все однозначные числа, сохраняя порядок, затем все двузначные числа, в обратном порядке.
2. Сформировать файл из символов и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все знаки препинания, затем все гласные, а затем все согласные, сохраняя исходный порядок в каждом случае

### Вариант 8

1. Сформировать файл из символов и за один просмотр файла напечатать элементы файла в следующем порядке: сначала все знаки препинания в исходном порядке, а затем все согласные в обратном порядке.
2. Сформировать файл из натуральных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, кратные 4, затем все нечетные числа, кратные 3, сохраняя исходный порядок в каждом случае.

### Вариант 9

1. В текстовом файле записана без ошибок формула вида: цифра или  $m$ (формула, формула), или  $p$ (формула, формула), где  $m(a, b) = (a+b) * 2$ ,  $p(a, b) = (a+b) \text{ div } 2$ . Вычислить значение данной формулы. Например:  $m(9, p(3, 5)) = 26$ .
2. Сформировать файл из вещественных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все отрицательные числа, а затем все положительные числа, сохраняя исходный порядок чисел в каждом случае.

### Вариант 10

1. Построить стек, элементами которого являются целые числа (числа взять из файла). Написать программу, которая находит сумму элементов, находящихся в стеке.
2. Сформировать файл из вещественных чисел и с помощью очереди за один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, кратные 5, а затем все остальные числа, сохраняя исходный порядок чисел в каждом случае.

## Лабораторная работа №12.

### Язык C/C++. Деки

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с деками.

**Указания.** Исходные данные поместить в файл input.txt. Результат – в файл output.txt.

#### Задание 1.

Написать программу, согласно варианту.

#### Задание 2.

Составить серию схем (см. материалы лекций), отражающих основные действия над списком в задаче (например, вставка элемента в определенное задачей место списка, удаление элемента, просмотр списка, перестановка элементов и т.д.).

Схемы должны в точности соответствовать реальной программе.

Порядок соединений и разрывов связей между элементами списка должен быть пронумерован и прокомментирован.

Для отчёта по лабораторной работе:

1. Работавшую программу с комментариями продемонстрировать преподавателю.
2. Листинг программы включить в отчет.

Задания оформить в электронном виде в формате doc, docx или pdf и файл прикрепить на учебные ресурсы (курс «Языки и методы программирования», задание 12: Лабораторная работа №12).

### Вариант 1

1.  $N$  натуральных чисел являются элементами двунаправленного списка  $L$ . Вычислить  $X_1 * X_n + X_2 * X_{n-1} + \dots + X_n * X_1$ .
2. Представить автотрассу в виде двунаправленного списка, элементы которого содержат информацию о названии населенных пунктов и расстоянии между ними. По заданным названиям двух населенных пунктов вычислить расстояние между ними.

### Вариант 2

1.  $N$  натуральных чисел являются элементами двунаправленного списка  $L$ . Вычислить  $(X_1 + X_n) * (X_2 + X_{n-1}) * \dots * (X_n + X_1)$ .
2. После игры в домино остался ряд камней. Используя двунаправленный список, определить, правильно ли расположены камни домино.

### Вариант 3

1. Считалка. Даны натуральные числа  $m$  и  $n$ . Предполагается, что  $n$  человек встают в круг и получают номера, считая против часовой стрелки, 1, 2, 3, ...,  $n$ . Затем, начиная с первого, также против часовой стрелки отсчитывается  $m$ -й человек (поскольку люди стоят по кругу, то за  $n$ -м человеком стоит первый). Этот человек выходит из круга, после чего, начиная со следующего, снова отсчитывается  $m$ -й человек и так до тех пор пока из всего круга не останется один человек. Определить его номер.
2. Пусть  $L$  обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая подсчитывает количество элементов списка  $L$ , у которых равные соседи.

### Вариант 4

1. Дан файл, содержащий некоторое количество слов. Вывести слова, являющиеся палиндромами (одинаково читается как слева направо, так и справа налево).

2. Пусть  $L$  обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая определяет, есть ли в списке  $L$  хотя бы один элемент, который равен следующему за ним по кругу элементу.

#### Вариант 5

1. Используя двунаправленный список, содержащий символы и их шифры, зашифровать текст, содержащийся в файле.

2. Пусть  $L$  обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая в списке  $L$  переставляет в обратном порядке все элементы между первым и последним вхождением элемента  $E$ , если  $E$  входит в  $L$  не менее двух раз.

#### Вариант 6

1. Создать двунаправленный список и разместить в нем случайным образом данные о колоде из 36 карт. Сократить список, убрав из него шестерки.

2. Дана запись многочлена (от переменной  $X$ ) произвольной степени с целыми коэффициентами, причем его одночлены могут быть и не упорядочены по степеням  $X$ , одночлены одной степени могут повторяться. Возможный пример

$$-8x^4 - 74x + 8x^4 + 5 - x^3$$

Требуется привести подобные члены в этом многочлене, после чего распечатать его по убыванию степеней  $X$ .

#### Вариант 7

1. Каталог книг, содержащихся в библиотеке, представлен в виде двунаправленного списка. Написать программу, обрабатывающую такой каталог (включение, исключение книг, поиск нужной книги).

2. Представить в виде двунаправленного списка схему автобусного маршрута. Запросить начальный и конечный пункты и определить направление движения для быстрого достижения цели.

#### Вариант 8

1. Последовательность камней домино представить в виде двунаправленного списка, заполненного произвольным образом. Исключить все дубли, вывести результат.

2. Пусть  $L$  обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая в конец непустого списка  $L$  добавляет все его элементы, располагая их в обратном порядке (например по списку из элементов 1, 2, 3 требуется построить список из элементов 1,2,3,3,2,1).

#### Вариант 9

1. Создать двунаправленный список и разместить в нем в определенном порядке данные о колоде игральных карт. Разработать программу, проводящую перемешивание колоды карт путем сдвига и перестановки ее частей. Результаты перемешивания вывести на печать.

2. Пусть  $L$  обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая в списке  $L$  удваивает каждое вхождение элемента  $E$ .

#### Вариант 10

1. Представить автотрассу в виде двунаправленного списка, элементы которого содержат информацию о названии населенных пунктов и расстоянии между ними. По заданному расстоянию между двумя населенными пунктами выбрать названия двух населенных пунктов, расстояние между которыми минимально отличается от заданного.

2. Пусть  $L$  обозначает кольцевой двунаправленный список с заглавным звеном. Описать функцию или процедуру, которая из списка  $L$ , содержащего не менее двух элементов, удаляет все элементы, у которых одинаковые "соседи" (первый и последний элементы считать соседями).

### Лабораторная работа №13.

#### Язык C/C++. Бинарные деревья

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с бинарными деревьями.

**Задание.** Задание №1 является общим для всех вариантов. К решению второй задачи имеет смысл приступать только после выполнения первого задания.

#### Задача 1. Общая.

Написать функцию удаления дерева, основываясь на материале лекции. Проверить работу функции на примере из лекции.

#### Задача 2.

#### Вариант 1

Составить программу поиска минимального ключа в дереве.

#### Вариант 2

Составить программу вычисления суммы всех ключей дерева

#### Вариант 3

Составить программу вычисления среднего арифметического всех ключей непустого дерева

#### Вариант 4

Составить программу, которая строит для дерева  $T$  его копию  $T1$ .

#### Вариант 5

Составить программу, которая определяет длину пути от корня дерева до вершины (узла) с ключом  $n$  (если  $n$  не входит в дерево, то длину пути считать равной  $-1$ )

#### Вариант 6

Составить программу вычисления суммы всех листьев дерева

#### Вариант 7

Составить программу, вычисляющую количество вершин дерева.

#### Вариант 8

Составить программу, увеличивающую все ключи дерева на число  $k$

#### Вариант 9

Составить программу поиска максимального ключа в дереве

#### Вариант 10

Составить программу поиска количества ключей в дереве, значения которых больше заданного числа  $a$ .

#### Лабораторная работа №14.

#### Язык C/C++. Элементы анализа эффективности алгоритмов

**ЦЕЛЬ РАБОТЫ:** Освоение принципов подсчета трудоемкости алгоритмов.

**Задание.** Для задачи по вариантам предыдущей лабораторной работы рассчитать трудоемкость алгоритма.

#### Модуль 4. Основы параллельного программирования.

#### Лабораторная работа №15.

#### Параллельное программирование на основе MPI. Операции над векторами

**ЦЕЛЬ РАБОТЫ:** Знакомство с кластером. Компиляция и запуск программ на кластере

**Задание.**

1. Зайдите на кластер под своим логином.
2. Загрузите Midnight Commander командой `mc`
3. Создайте папку для хранения программ клавишей F7, перейдите в созданную папку.
4. Загрузите в созданную папку исходный код программы с помощью WinSCP или создайте новый файл в Midnight Commander, нажав Shift+F4 и отредактировав его в текстовом редакторе `mc` (переход в текстовый редактор F4)
5. Наберите текст программы, выводящей сообщения (Hello, world!!) от процессов (пример из лекции 16). Для сохранения изменений в текстовом редакторе `mc` используйте F2.
6. Для компиляции программы:
  - a. Сверните панели `mc` (`ctrl+o`)
  - b. В командной строке наберите **module avail**. Вы увидите список установленных в системе программных модулей и компиляторов. Нам нужен **openmpi/gcc/64/1.4.5**
  - c. Загрузите модуль командой: **module load openmpi/gcc/64/1.4.5**
  - d. Выполните компиляцию файла программы командой: **mpicc -o <имя\_объектного\_файла> <имя\_компилируемого\_файла>**. Если компиляция выполнена без ошибок, то в папке появится объектный файл (выделен зеленым цветом)
  - e. Запустите программу на выполнение командой: **mpirun -np <количество\_процессов> <имя\_объектного\_файла>**
7. Изменить предыдущую программу так, чтобы сообщения от процессоров выводились в порядке их нумерации (рангов).
8. Написать программу, которая передаёт некоторое число по кольцу: от процесса с рангом 0, на процесс 1, процесс 1 прибавляет к числу 1 и передаёт на процесс 2 и т.д. Полученное в результате число выводится на консоль процессом с рангом 0. При написании программы:
  - a. Использовать стандартный режим передачи сообщений
  - b. Добавить расчёт времени работы программы, зафиксировать время работы программы на 20 процессах.
  - c. Изменить режим передачи данных на буферизованный и зафиксировать время работы программы на 20 процессах.
  - d. Изменить режим передачи данных на режим по готовности и зафиксировать время работы программы на 20 процессах
  - e. Изменить режим передачи данных на синхронный режим и зафиксировать время работы программы на 20 процессах
  - f. Определить наиболее быстрый и наиболее медленный режимы работы программы.

#### Лабораторная работа №16.

#### Параллельное программирование на основе MPI. Парные операции передачи данных.

**ЦЕЛЬ РАБОТЫ:** Продолжение знакомства с кластером. Отладка готовой программы и проведение серии вычислительных экспериментов

### Задание 1.

1. Написать программу передачи числа по кольцу так, чтобы четные процессы передавали числу четным процессам, а нечетные - нечетным. При написании программы:
  - a. Использовать стандартный режим передачи сообщений
  - b. Добавить расчёт времени работы программы, зафиксировать время работы программы на 20 процессах.
2. Изменить режим передачи данных на буферизованный и зафиксировать время работы программы на 20 процессах.
3. Изменить режим передачи данных на режим по готовности и зафиксировать время работы программы на 20 процессах
4. Изменить режим передачи данных на синхронный режим и зафиксировать время работы программы на 20 процессах
5. Определить наиболее быстрый и наиболее медленный режимы работы программы.

### Задание 2.

1. Откомпилировать и запустить на кластере задачу суммирования элементов вектора, в которой:
  - a. должна быть дописана функция заполнения массива (случайными числами или согласно какой-либо математической формуле);
  - b. должен быть добавлен расчет времени выполнения программы
2. Запустить программу с одной и той же размерностью вектора несколько раз на 2-х, 4-х, 8-ми, 16-ти и 32-х процессорах с фиксацией времени выполнения программы при каждом запуске.
3. Построить в Excel (или Calc) график зависимости скорости выполнения программы от количества процессоров.
4. Запустить программу несколько раз на 16 процессорах, изменяя размерность массива в несколько раз при каждом запуске (100, 1000, 10000 и 100000 элементов). Зафиксировать время выполнения
5. Построить в Excel (или Calc) график зависимости скорости выполнения программы от увеличения количества входных данных при одном и том же количестве процессоров

### Лабораторная работа №17.

#### Параллельное программирование на основе MPI. Коллективные операции передачи данных. Редукция.

**ЦЕЛЬ РАБОТЫ:** Продолжение знакомства с кластером. Разработка программ с применением операций широковещательной рассылки и редукции данных

### Задание 1.

Изменить задачу о суммировании элементов массива так, чтобы в ней использовались операция редукции данных и операция широковещательной рассылки.

А) Измерить время работы программы без редукции и Vcast() на массиве размерностью 10000 элементов на 2, 4, 8 и 16 процессах. Построить график изменения скорости вычислений в зависимости от количества процессоров

Б) Измерить время работы программы с использованием редукции и Vcast() на массиве размерностью 10000 элементов на 2, 4, 8 и 16 процессах. Построить график изменения скорости вычислений в зависимости от количества процессоров.

В) Сравнить два графика и сделать вывод.

### Задание 2.

**При написании программы использовать операции редукции (если возможно) и широковещательной рассылки.**

#### Вариант 1

Составить параллельную программу поиска минимального элемента вектора.

#### Вариант 2

Составить параллельную программу поиска максимального элемента вектора.

#### Вариант 3

Составить параллельную программу вычисления среднего арифметического элементов вектора

#### Вариант 4

Составить параллельную программу подсчета количества четных элементов вектора.

#### Вариант 5

Составить параллельную программу подсчета количества нечетных элементов вектора

#### Вариант 6

Составить параллельную программу подсчета суммы нечетных элементов вектора

#### Вариант 7

Составить параллельную программу подсчета суммы четных элементов вектора.

#### Вариант 8

Составить параллельную программу, увеличивающую все элементы вектора на число k

#### Вариант 9

Составить параллельную программу, увеличивающую все элементы вектора в k раз

#### Вариант 10

Составить параллельную программу подсчёта среднего арифметического нечётных элементов вектора

#### Лабораторная работа №18.

##### Параллельное программирование на основе MPI. Коллективные операции передачи данных. Функции MPI\_Scatter и MPI\_Gather.

**ЦЕЛЬ РАБОТЫ:** Продолжение знакомства с кластером. Разработка программ с применением операций обобщенного сбора данных и распределения данных

##### Задание 1.

1. Реализовать задачу умножения матрицы на вектор, рассмотренную на лекции.
2. Замерить время работы программы на 2, 4, 8 и 16-ти процессах. Построить график зависимости времени работы программы от количества процессов. Для замеров размерность матрицы выбирать так, чтобы на каждый процесс приходилось не менее 5-10 строк, в каждой из которых не менее 100 элементов.

##### Задание 2.

Написать программу по заданному алгоритму:

А) На нулевом процессе создать одномерный массив размерностью  $100 \cdot N$ , где  $N$  – число процессов программы. С помощью соответствующей операции раздать части массива на все процессы. В своей части массива все процессы должны найти среднее арифметическое и разделить на него все элементы своей части массива. Затем результаты работы собрать на нулевом процессе.

Б) для проверки вычислений на экран вывести 10 первых и 10 последних элементов исходного массива до начала обработки и после всех расчетов

В) замерить время работы программы на 2, 4, 8 и 16-ти процессах. Построить график зависимости времени работы программы от количества процессов

#### Лабораторная работа №19.

##### Параллельное программирование на основе MPI. Решение СЛУ методом Гаусса.

**ЦЕЛЬ РАБОТЫ:** Продолжение знакомства с кластером. Разработка программ с применением операций коллективных операций передачи данных

##### Задание 1.

1. Реализовать задачу решения системы линейных уравнений методом Гаусса, рассмотренную на лекции.

Замерить время работы программы на 2, 4, 8 и 16-ти процессах. Построить график зависимости времени работы программы от количества процессов. Для замеров размерность матрицы выбирать так, чтобы на каждый процесс приходилось не менее 5-10 строк, в каждой из которых не менее 100 элементов.

#### Модуль 5. Основы объектно-ориентированного программирования.

##### Лабораторная работа №20.

##### Язык C++. ООП. Создание класса

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с классами, знакомство со способами описания свойств и методов класса.

##### Задание 1. Общее.

Дописать недостающие методы класса «Complex», рассмотренного на лекции:

1. Методы вычитания, умножения и деления комплексных чисел
2. Метод перевода комплексного числа в тригонометрическую форму

##### Задание 2.

1. Объявить класс в соответствии с вариантом и определить для него 2 конструктора.
2. Определить функции-методы класса для ввода и вывода свойств внутри объявления класса.
3. Составить программу, которая определяет три объекта класса и выводит их на экран. Первый объект должен инициализироваться по умолчанию, второй использовать 1-й конструктор, третий - функцию ввода данных.
4. Выполнить программу и проверить соответствие заданных и полученных данных.
5. Определить функции ввода и вывода вне объявления класса и повторить выполнение программы.

##### Вариант 1

Класс "Рациональная дробь" с данными: числитель и знаменатель.

##### Вариант 2

Класс "Вектор на плоскости" с данными: проекция вектора на оси X и Y.

##### Вариант 3

Класс "Змейка" с данными: длина, положение головы (x; y).

##### Вариант 4

Класс "Окружность" с данными центр и радиус окружности.

##### Вариант 5

Класс "Номер телефона" с данными код города, код станции и номер линии.



### Вариант 6

Класс "Квадрат" с данными центр квадрата и его сторона.

### Вариант 7

Класс "Прямоугольник" с данными верхняя левая и правая нижняя точки.

### Вариант 8

Класс "Вектор на плоскости" с данными модуль вектора и угол между вектором и положительным направлением оси X.

### Вариант 9

Класс "Дата" с данными число, номер месяца и две последние цифры года.

### Вариант 10

Класс "Время" с данными часы, минуты и секунды.

## Лабораторная работа №21-22.

### Язык C/C++. Конструкторы и деструкторы. Наследование

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с классами, знакомство со способами описания потомков класса.

#### Задание 1. Общее

1. Для класса «Complex» описать конструкторы:

- А) по умолчанию
- Б) с параметрами
- В) копирования

2. Для класса «Complex» описать метод печати комплексного числа как дружественную функцию

Продемонстрировать работу всех методов и конструкторов

#### Задание 2. По вариантам

1. Для задачи предыдущей лабораторной объявить функции ввода и вывода как "друзей" класса с параметром - объектом класса, передаваемым по значению. Изменить определение этих функций и повторить выполнение программы.

2. Для задачи предыдущей лабораторной определить все три вида конструкторов: по умолчанию, с параметрами, копирования.

3. Для задачи предыдущей лабораторной создать класс-потомок согласно заданию. В классе-потомке определить дополнительные свойства (согласно заданию) и методы (возможно переопределение методов класса-родителя). Также определить в потомке все три вида конструкторов.

4. Проверить работу всех функций нового класса (как унаследованных, так и собственных).

### Вариант 1

1. Класс "Рациональная дробь" с данными: числитель и знаменатель. Определить метод перевода рациональной дроби в десятичную. Определить класс-потомок: дробь, в которой числитель и знаменатель рациональные дроби (для некоторых школьных алгебраических задач). Определить в потомке функцию, переводящую «четырёхэтажную» дробь в обычную.

### Вариант 2

1. Класс "Вектор на плоскости" с данными проекция вектора на оси X и Y. Определить метод, вычисляющий длину вектора. Определить класс-потомок: вектор в пространстве с дополнительным свойством – проекция на ось Z. Определить в потомке метод, вычисляющий угол между вектором и осью OX.

### Вариант 3

1. Класс "Змейка" с данными: длина, положение головы (x; y). Определить метод, который увеличивает длину змейки на один, если съедена «хорошая» пища. Определить класс-потомок: ящерица с дополнительными данными: положение хвоста (tailX, tailY) и длина хвоста. Определить в потомке функцию «отбросить хвост»

### Вариант 4

1. Класс "Окружность" с данными центр и радиус окружности. Описать метод вычисляющий площадь кольца, полученного при наложении двух окружностей. Определить класс-потомок: эллипс с дополнительным свойством – значение второго радиуса. Определить в потомке метод, вычисляющий эксцентриситет эллипса.

### Вариант 5

1. Класс "Номер телефона" с данными код города, код станции и номер линии. Добавить свойство, определяющее занята линия или нет, описать метод выбирающий свободные линии из заданного множества (трёх телефонов). Определить класс-потомок: мини АТС с дополнительным свойством – количество номеров на АТС. Определить логическую функцию, определяющую есть ли свободные номера (не закреплённые за абонентом).

### Вариант 6

1. Класс "Квадрат" с данными центр квадрата и его сторона. Определить метод, вычисляющий длину диагонали квадрата. Определить класс-потомок: прямоугольник с дополнительным свойством – значение ширины. Определить в потомке метод, вычисляющий площадь прямоугольника.

### Вариант 7

1. Класс "Прямоугольник" с данными верхняя левая и правая нижняя точки. Описать метод, определяющий длины сторон. Определить класс-потомок: трапеция с дополнительными свойствами – координаты оставшихся

двух точек. Определить в потомке метод, определяющий, образует ли фигура, построенная по четырём точкам, трапецию.

#### Вариант 8

1. Класс "Вектор на плоскости" с данными модуль вектора и угол между вектором и положительным направлением оси X. Описать метод, вычисляющий координаты вектора. Определить класс-потомок: вектор в пространстве с дополнительным свойством – угол между вектором и положительным направлением оси Y. Определить в потомке метод, вычисляющую, является ли вектор параллельным оси Z.

#### Вариант 9

1. Класс "Дата" с данными число, номер месяца и две последние цифры года. Описать метод, вычисляющий количество дней до нового года. Заданный день не учитывается. Определить класс-потомок: период с дополнительными свойствами – число, номер месяца и две последние цифры года второй даты. Определить в потомке метод, определяющий, период в днях, прошедший между датами (для упрощения високосные годы не учитывать).

#### Вариант 10

1. Класс "Время" с данными часы, минуты и секунды. Определить метод, вычисляющий опоздали вы в кинотеатр на фильм или нет. Если опоздали, то на сколько минут, если успеваете, то сколько осталось минут в запасе. Определить класс-потомок: период с дополнительными свойствами – часы, минуты и секунды второй временной точки. Определить в потомке метод, определяющий, период в минутах, прошедший между временными точками (считать, что обе точки находятся в пределах одних суток).

#### Лабораторная работа №23.

#### Язык C/C++. Полиморфизм. Перегрузка операций

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с классами, знакомство со способами описания свойств и методов класса, знакомство с перегрузкой операций.

#### Задание 1. Общее

- Для класса «Complex» перегрузить операции:
  - присваивания,
  - сложения,
  - умножения
  - при необходимости, деления,
  - сравнения дробей (<, >, ==).
- Проверить правильность работы перегруженных операций.

#### Задание 2. По вариантам.

- Для задачи предыдущей лабораторной проверить наличие конструкторов. Перегрузить для своего класса операции присваивания, сложения, умножения и, при необходимости, деления, а также сравнения (<, >, ==).
- Проверить правильность работы перегруженных операций.
- Разработать дополнительную функцию для базового и производного классов согласно варианту и проверить их работу.

#### Вариант 1

1. Класс "Рациональная дробь" с данными: числитель и знаменатель. Определить метод, выделяющий целую часть неправильной дроби.

#### Вариант 2

1. Класс "Вектор на плоскости" с данными проекция вектора на оси X и Y. Определить метод, вычисляющий длину вектора.

#### Вариант 3

1. Класс "Змейка" с данными: длина, положение головы (x; y). Определить метод, вычисляющий расстояние до ближайшей стены (параметрами функции взять длину и ширину игрового поля). Считать, что левый нижний угол игрового поля находится в начале координат.

#### Вариант 4

1. Класс "Окружность" с данными центр и радиус окружности. Описать метод вычисляющий площадь кольца, полученного при наложении двух окружностей.

#### Вариант 5

1. Класс "Номер телефона" с данными код города, код станции и номер линии. Добавить свойство, определяющее занята линия или нет, описать метод выбирающий свободные линии из заданного множества (трёх телефонов).

#### Вариант 6

1. Класс "Квадрат" с данными центр квадрата и его сторона. Определить метод, вычисляющий длину диагонали квадрата.

#### Вариант 7

1. Класс "Прямоугольник" с данными верхняя левая и правая нижняя точки. Описать метод, определяющий длины сторон.

### Вариант 8

1. Класс "Вектор на плоскости" с данными модуль вектора и угол между вектором и положительным направлением оси X. Описать метод, вычисляющий координаты вектора.

### Вариант 9

1. Класс "Дата" с данными число, номер месяца и две последние цифры года. Описать метод, вычисляющий количество дней до нового года. Заданный день не учитывается.

### Вариант 10

1. Класс "Время" с данными часы, минуты и секунды. Определить метод, вычисляющий опоздали вы в кинотеатр на фильм или нет. Если опоздали, то на сколько минут, если успеваете, то сколько осталось минут в запасе.

## Лабораторная работа №24.

### Язык C/C++. Поточковый ввод и вывод данных

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с классами, знакомство с организацией ввода-вывода данных с использованием классов C++.

#### Задание 1.

Для класса «Complex»:

- 1) организовать ввод данных для базового и производного классов из файла с использованием потокового ввода.
- 2) проверить правильность работы метода (функции) ввода данных.
- 3) организовать вывод результатов для базового и производного классов в файл с использованием потокового вывода. При организации вывода использовать форматирование вывода с помощью флагов или функций.
- 4) проверить правильность работы вывода данных для базового и производного классов.

#### Задание 2.

Для задачи предыдущей лабораторной:

- 1) организовать ввод данных для базового и производного классов из файла с использованием потокового ввода.
- 2) проверить правильность работы метода (функции) ввода данных.
- 3) организовать вывод результатов для базового и производного классов в файл с использованием потокового вывода. При организации вывода использовать форматирование вывода с помощью флагов или функций.
- 4) проверить правильность работы вывода данных для базового и производного классов.
- 5) запрограммировать возможность выбора пользователем способа ввода и вывода: с клавиатуры или через файл.

### Вариант 1

Класс "Рациональная дробь" с данными: числитель и знаменатель.

### Вариант 2

Класс "Вектор на плоскости" с данными: проекция вектора на оси X и Y.

### Вариант 3

Класс "Змейка" с данными: длина, положение головы (x; y).

### Вариант 4

Класс "Окружность" с данными центр и радиус окружности.

### Вариант 5

Класс "Номер телефона" с данными код города, код станции и номер линии.

### Вариант 6

Класс "Квадрат" с данными центр квадрата и его сторона.

### Вариант 7

Класс "Прямоугольник" с данными верхняя левая и правая нижняя точки.

### Вариант 8

Класс "Вектор на плоскости" с данными модуль вектора и угол между вектором и положительным направлением оси X.

### Вариант 9

Класс "Дата" с данными число, номер месяца и две последние цифры года.

### Вариант 10

Класс "Время" с данными часы, минуты и секунды.

## Лабораторная работа №25.

### Язык C/C++. Шаблоны классов и функций

**ЦЕЛЬ РАБОТЫ:** Освоение принципов работы с шаблонами классов и функций

#### Задание 1.

Написать класс «Квадратная матрица», в котором реализовать:

1) Поля: одномерный динамический массив, размера  $n \times n$ , где  $n$  – размер матрицы (размер матрицы – это второе поле соответственно). Для реализации вспомнить параллельное программирование в части представления двумерной матрицы в виде длинного одномерного массива.

2) Методы:

- a. Умножение матрицы на число (в том числе дробное)
- b. Транспонирование матрицы
- c. Сложение двух матриц
- d. Умножение матрицы на вектор (вектор можно рассматривать как такую же матрицу размера  $n \times 1$ )
- e. Умножение двух матриц
- f. Переопределить оператор присваивания.

Проверить правильность работы методов класса в функции main()

### Задание 2.

Преобразовать написанный класс «Квадратная матрица» в шаблон класса. Применить шаблон к матрицам с данными типов int, long, double

### Задание 3.

Преобразовать в шаблон класс, написанный в ходе предыдущих лабораторных согласно варианту.

#### Вариант 1

Класс "Рациональная дробь" с данными: числитель и знаменатель.

#### Вариант 2

Класс "Вектор на плоскости" с данными проекция вектора на оси X и Y.

#### Вариант 3

Класс "Змейка" с данными: длина, положение головы (x; y).

#### Вариант 4

Класс "Окружность" с данными центр и радиус окружности.

#### Вариант 5

Класс "Номер телефона" с данными код города, код станции и номер линии.

#### Вариант 6

Класс "Квадрат" с данными центр квадрата и его сторона.

#### Вариант 7

Класс "Прямоугольник" с данными верхняя левая и правая нижняя точки.

#### Вариант 8

Класс "Вектор на плоскости" с данными модуль вектора и угол между вектором и положительным направлением оси X.

#### Вариант 9

Класс "Дата" с данными число, номер месяца и две последние цифры года.

#### Вариант 10

Класс "Время" с данными часы, минуты и секунды.

## Модуль 6. Основы функционального программирования.

### Лабораторная работа №26.

#### Лисп. Основные конструкции языка.

**ЦЕЛЬ РАБОТЫ:** Знакомство со средой программирования clisp и диалектом Common lisp. Практическое использование основных конструкций языка.

#### Задание

##### Часть 1. Общая.

1. Решить следующие задачи:

- a. Составить программу вычисления значения функции. Значение  $x$  должно вводиться с клавиатуры.  $f(x) = \cos^2 x^3 + (\sin x + 1)^2$ .
- b. Из списков (a b c) и (x y z) получить список (a y z)
- c. Составить программу поиска предпоследнего элемента произвольного списка.

##### Часть 2. По вариантам.

#### Вариант 1

Написать программу, которая переводит температуру в градусах по Фаренгейту в градусы Цельсия по формуле:

$$C = \frac{5}{9}(F - 32)$$

#### Вариант 2

Найти площадь кольца по заданным внешнему и внутреннему радиусам.

#### **Вариант 3**

Составить программу вычисления значения функции  $y = 12x^2 + 7x - 16$  при любом значении  $x$ .

#### **Вариант 4**

Даны катеты прямоугольного треугольника. Найти гипотенузу.

#### **Вариант 5**

Посчитать значение математического выражения для допустимых значений  $x$ :

$$y = \frac{5}{9} \sqrt{\frac{\cos(x)}{\sin(x)}} - \frac{1}{x^3}$$

#### **Вариант 6**

Даны стороны треугольника. Найти площадь треугольника.

#### **Вариант 7**

Составить программу вычисления значения функции при допустимом значении  $x$ .

$$y = \sin \frac{3.2 + \sqrt{1+x}}{|5x|}$$

#### **Вариант 8**

Даны длины сторон прямоугольника. Найти его периметр и длину диагонали.

#### **Вариант 9**

Напишите программу, которая переводит сантиметры в дюймы.

#### **Вариант 10**

Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.

### **Лабораторная работа №27.**

#### **Лисп. Среда clisp. Основные конструкции языка. Ветвления. Циклы**

**ЦЕЛЬ РАБОТЫ:** Знакомство со средой программирования clisp и диалектом Common lisp. Практическое использование основных конструкций языка. Организация ветвлений с помощью функции cond. Использование именованных и безымянных функций

**Задание.**

**Указание:** для решения задач использовать именованные функции

#### **Вариант 1**

1. Даны длины сторон двух прямоугольников. Определить, какой из прямоугольников больше по площади. Для определения площади прямоугольника написать именованную функцию.
2. Заменить в заданном математическом предложении все вхождения sin на cos и sqrt на abs.
3. Найти все слова, которые есть в первом предложении и отсутствуют во втором.

#### **Вариант 2**

1. Известны даты рождения двух человек. Определить, кто из них старше. Для решения задачи написать именованную функцию
2. Для заданного предложения определить максимальное, содержащееся в тексте число.
3. Дан текст из латинских букв и знаков препинания. Составить программу частотного анализа букв этого текста, т.е. напечатать каждую букву с указанием количества ее вхождений и процента вхождений.

#### **Вариант 3**

1. Даны два списка. Выяснить, равны ли их последние элементы. Для поиска последнего элемента списка написать именованную функцию
2. Напечатать все слова из заданного предложения, состоящие из тех же букв, что и первое слово предложения.
3. Дана последовательность чисел. Определить сколько из них входят в последовательность Фибоначчи.

#### **Вариант 4**

1. Даны катеты двух треугольников. Написать программу, определяющую, можно ли из треугольников сложить прямоугольник. При написании программы использовать собственную именованную функцию
2. Слова в предложении разделены пробелами. Упорядочить слова в соответствии с ростом их длины.
3. Определить в заданной строке количество знаков арифметических действий (+, -, и т.д.).

#### **Вариант 5**

1. Написать именованную функцию, которая по двум заданным спискам и заданному элементу удаляет этот элемент из обоих списков.
2. Из заданного предложения выбрать только те символы, которые встречаются в нем только один раз, в том порядке, как они расположены в тексте.
3. Дан текст, состоящий из латинских букв и знаков препинания. Определить каких букв, гласных или согласных, больше и на сколько.

### **Вариант 6**

1. Написать именованную функцию, определяющую является ли первый элемент заданного списка меньше последнего.
2. Определить число вхождений заданного слова в текст. Слова в тексте разделены пробелами.
3. В возрастающем порядке напечатать все числа из интервала от 1 до 250, представимые в виде:  $n^2 + m^2$ , где n и m – целые числа.

### **Вариант 7**

1. Даны площади и основания двух равнобедренных трапеций. Определить, какая из трапеций имеет большую высоту. Для определения высоты трапеций написать именованную функцию.
2. Найти в заданном тексте самое короткое слово.
3. Найти количество различных цифр в десятичной записи произвольного числа.

### **Вариант 8**

1. Даны два списка. Выяснить, равны ли их четвертые элементы. Для поиска четвертого элемента списка написать именованную функцию.
2. Найти все слова, общие для двух предложений
3. Найти и напечатать в порядке убывания все простые числа из диапазона от 2 до 201.

### **Вариант 9**

1. Известны даты рождения двух человек. Определить, кто из них младше. Для решения задачи написать именованную функцию.
2. Для заданного предложения определить минимальное, содержащееся в тексте число.
3. Дан текст, состоящий из латинских букв. Найти те буквы, которые входят в текст не менее 2-х раз.

### **Вариант 10**

1. Даны радиусы двух окружностей. Определить, какая из окружностей имеет меньшую длину окружности. Для определения длины окружности написать именованную функцию.
2. Найти в заданном тексте самое длинное слово.
3. В заданном тексте слова разделены пробелами. Удалить лишние пробелы так, чтобы между словами осталось только по одному пробелу.

## **Лабораторная работа №28.**

### **Лисп. Циклы и конструкция LOOP. Параметры функций.**

**ЦЕЛЬ РАБОТЫ:** Применение различных типов параметров при написании пользовательских функций на Лисп.

#### **Задание**

*В каждом варианте для решения задачи использовать функцию. Реализовать функцию в четырёх вариантах с использованием обязательных, необязательных, остаточных, именованных параметров. **Исходные данные прочитать из файла.***

#### **Вариант 1**

Прочитать из файла значения переменных x и y, а также список l. Написать рекурсивную функцию, заменяющую в списке l первое вхождение x на y. Список задать при вызове функции.

#### **Вариант 2**

Удалить из списка все элементы, меньшие заданного x. Например: дано (3 6 8 3 5 6), x=5, получить (6 8 6)

#### **Вариант 3**

Прочитать из файла значение переменной x. Написать рекурсивную функцию, удаляющую все вхождения x из списка l. Список задать при вызове функции, содержимое списка прочитать из файла.

#### **Вариант 4**

Написать рекурсивную функцию инвертирования списка. Список задать при вызове функции. Стандартную функцию инвертирования не использовать.

#### **Вариант 5**

Задать с клавиатуры значение переменной x. Написать рекурсивную функцию, определяющую, является ли x элементом списка l. Список задать при вызове функции.

#### **Вариант 6**

Прочитать из файла значения переменных x и y. Написать рекурсивную функцию, заменяющую в списке l все вхождения x на y. Список задать при вызове функции.

#### **Вариант 7**

Написать рекурсивную функцию, объединяющую два списка. Список задать при вызове функции. Стандартную функцию объединения не использовать.

#### **Вариант 8**

Прочитать из файла значения переменных x и y. Написать рекурсивную функцию, заменяющую в списке l последнее вхождение x на y. Список задать при вызове функции.

#### **Вариант 9**

Удалить из списка все элементы, большие заданного x. Например: дано (3 6 8 3 5 6), x=5, получить (3 3 5)

### Вариант 10

Прочитать из файла значение переменной  $x$ . Написать рекурсивную функцию, удаляющую  $x$  (последнее вхождение) из списка  $l$ . Список задать при вызове функции.

## Модуль 7. Основы логического программирования.

### Лабораторная работа №29.

#### Visual Prolog. Знакомство.

**ЦЕЛЬ РАБОТЫ:** Ознакомление со средой Visual Prolog; составление и выполнение простейших программ.

#### Задание

#### Варианты заданий

##### Вариант 1.

1. **Задача о симпатиях.** Васе нравится мороженое. Васе нравится кино. Васе нравится спать. Пете нравится футбол. Пете нравится Маша.

a. Найти ответы на вопросы:

i Нравится ли Васе кино?

ii Нравится ли Васе футбол?

iii Нравится ли Васе Маша?

iv Что нравится Васе?

b. Добавить правило «Васе нравится всё, что нравится Пете» и найти ответы на вопросы:

i Нравится ли Васе футбол?

ii Что нравится Васе?

2. **Задача о Сократе (с одноместными предикатами).**

a. Известно, что Сократ человек. Известно, что все люди смертны. Найти ответ на вопрос «Сократ смертен?».

b. Известно, что Сократ, Вася и Вова люди, и все люди смертны. Найти имена смертных.

##### Вариант 2.

1. **Задача о братьях.** Известно, что Адам отец Каина и Авеля, Ева - мать Каина и Авеля. Отец является родителем, мать тоже является родителем. Если у двоих людей один родитель, то они братья. Написать Пролог-программу, которая делает вывод, что Каин и Авель – братья.

2. **Задача о мышах (с одноместными предикатами).**

a. Известно, что мыши - животные. Известно, что все животные не говорят. Найти ответ на вопрос «Мыши говорят?».

b. Известно, что мыши, коты и крысы - животные, и все животные не говорят. Найти названия животных.

##### Вариант 3.

1. **Задача о соседе.**

Известно, что сосед был убит у себя дома в 12 часов, к нему заходили почтальон в 12 часов и сантехник в 10 часов. Убийца тот, кто заходил в дом в час убийства. Написать Пролог-программу, которая делает вывод, кто является убийцей соседа.

Указание:

для создания факта о времени убийства соседа объявить предикат `was_killed(time)`; правило определения убийцы перефразировать так: некто является убийцей жертвы в том случае, если он находился в доме в то же время, когда была убита жертва.

2. Шмык и Дрыг – угубки. Если почесать угубку за ухом, он начнет довольно шипеть. Если угубок довольно зашипит, то молоко поблизости скиснет. Прык довольно зашипел и молоко скисло. Шмока почесали за ухом.

Ответить на вопросы:

a. Кто угубок?

b. Угубок ли Прык?

c. Угубок ли Шмок?

##### Вариант 4.

1. **Задача о покушении на Цезаря.**

Можно ли из следующей совокупности утверждений:

(F1) Марк был римлянином

(F2) Цезарь был диктатором

(F3) Те римляне, которые ненавидели диктатора, пытались убить его

(F4) Римляне либо были преданы диктатору, либо ненавидели его

(F5) Марк не был предан Цезарю

вывести доказательство того, что Марк пытался убить Цезаря?

Указание:

- для каждого из подчеркнутых отношений/свойств создать предикат (их будет 6)

- будет 5 одноместных предикатов на свойства и один двухместный на фразу «они пытались убить его» - *try\_kill(symbol кто\_пытался\_убить, symbol кого\_пытались\_убить)*
- утверждениям F1, F2, F5 соответствуют факты, F3 – правило, F4 – два правила

#### Вариант 5.

##### 1. Задача о соседе

Известно, что сосед был убит у себя дома в 12 часов, а его кошка в 11 часов, к нему заходили почтальон в 12 часов и сантехник в 10 часов. Убийца тот, кто заходил в дом в час убийства. Написать Пролог-программу, которая делает вывод, кто является убийцей соседа.

Указание:

для создания фактов об убийствах объявить предикат *was\_killed(zhertva, time)*; правило определения убийцы перефразировать так: некто является убийцей жертвы в том случае, если он находился в доме в то же время, когда была убита жертва.)

2. Люди живут в цветных домах. Те, кто живет в синих домах – спокойные, те, кто живет в красных – агрессивные, те, кто живет в зеленых – добрые. Петя и Ира живут в синем доме, Сара – добрая, Коля живет в красном доме. Кто является спокойным? Кто живет в красном доме? В зеленом?

#### Вариант 6.

1. Цветы - это зеленые звери. Цветы пьют морсик. Сеня – зеленый зверь, Кеша пьет морсик, Соня морсик не пьет. Кто является цветком? Кто не является цветком?

2. **Задача о вороне и питоне.** Птица откладывает яйца и имеет крылья. Рептилии откладывают яйца и имеют чешую. Ворона откладывает яйца. Питон откладывает яйца. Питон имеет чешую. Ворона имеет крылья. Является ли ворона птицей?

a. Описать в программе два объекта (обозначаемых константами) и пять свойств (обозначаемых одноместными предикатами)

b. Описать в программе пять феноменов (обозначаемых константами) и два отношения (обозначаемых двухместными предикатами)

#### Вариант 7.

1. Тони, Майкл и Джон – члены альпинистского клуба. Каждый член клуба или горнолыжник или скалолаз, или и то и другое. Никто из скалолазов не любит дождь. Все горнолыжники любят снег. Майкл любит все, что не любит Тони, и не любит все, что любит Тони. Тони любит снег и дождь. Есть ли член клуба, который является скалолазом и не является горнолыжником? И кто он?

2. Все улитки являются горами. Все горы любят кошек. Значит, все улитки любят кошек. Мел и Фер – любят кошек, Зиг кошек не любит. Кто является улитками? Кто является горами?

#### Вариант 8.

1. **Задача о любимых фруктах.** Мэри любит яблоки, персики и ананасы. Кати любит все то, что любит Мэри и еще бананы. Ответить на вопросы:

- Любит ли Мэри бананы?
- Любит ли Мэри яблоки?
- Что любит Кати?
- Любит ли Кати кукурузу?

2. Анна, Лена и Анжела посещают кружок вышивания. Каждый член кружка умеет вышивать или крестиком или атласной гладью или и то и другое. Лена умеет вышивать крестиком и не умеет гладью, Анжела умеет то, что не умеет Лена. Вера вышивает двусторонней гладью, а Соня вяжет. Является ли Вера членом кружка? Что умеет делать Анжела?

#### Вариант 9.

1. **Задача о любимых играх.** Вова любит Call of Duty, HalfLife и не любит Witcher и Neverwinter. Вера любит то, что не любит Вова. Ответить на вопросы:

- Что любит Вера?
- Любит ли Вова Allods?
- Любит ли Вера Call of Duty?
- Любит ли Вова HalfLife?

2. Все цветы имеют лепестки и стебель. Все травы имеют листья и стебель. Бамбук имеет стебель. Маргаритка имеет стебель и лепестки. Бамбук имеет листья. Является ли Бамбук травой? Является ли маргаритка травой?

#### Вариант 10.

1. Известно, что грымзики или полосатые или рогатые или и то и другое вместе. Грюндик не имеет рогов, Грюмник не имеет рогов и не полосат. Ответить на вопросы:

- Является ли Грюмник грымзиком?
- Является ли Грюндик грымзиком?
- Кто является грымзиком?
- Кто не является грымзиком?



2. Все реки впадают в море или в другие реки. Если река впадает в море, то и ее притоки впадают в это море. Енисей впадает в Карское море. Ангара и Подкаменная Тунгуска впадают в Енисей. Иртыш не впадает в Енисей. Что впадает в Карское Море?

### **Лабораторная работа №30.**

#### **Visual Prolog. Списки, знакомство со списками, решение простых задач.**

**ЦЕЛЬ РАБОТЫ:** Закрепление знаний о языке пролог; составление рекурсивных программ обработки списков.

#### **Задание**

#### **Варианты заданий**

##### **Вариант 1.**

1. Написать программу поиска максимального элемента списка.
2. В списке канцелярских товаров найти наименование, следующее за наименованием «тетрадь 12 листов» и удалить все его вхождения.

##### **Вариант 2.**

1. Написать программу поиска минимального элемента списка
2. Даны 2 списка. Удалить из второго все вхождения головы первого списка.

##### **Вариант 3.**

1. Написать программу удаления всех вхождений элемента X из списка.
2. Дано 2 списка. Инвертировать первый список, во втором удалить последний элемент. Результаты объединить.

##### **Вариант 4.**

1. Решить задачу обращения списка. Например: дан список [4, 6, 12, 5], в результате работы программы получается список: [5, 12, 6, 4]
2. Дан список. Найти максимальный элемент хвоста, длину хвоста и обратить хвост.

##### **Вариант 5.**

1. Написать программу, объединяющую 2 списка и удаляющую из общего списка все вхождения головы второго списка.
2. Добавить к списку товаров на некотором складе наименование «Сапоги». Объединить список товаров со списком цен. Выяснить есть ли товар «Шляпка», если есть, определить его цену.

##### **Вариант 6.**

1. Удалить из списка заданный элемент, найти длину оставшейся части списка.
2. Объединить два списка, найти последний элемент, проверить, является ли он минимальным.

##### **Вариант 7.**

1. Подсчитать количество вхождений первого элемента в список.
2. Из первого списка исключить первый, из второго – последний элементы. Списки объединить, найти длину результата.

##### **Вариант 8.**

1. Написать программу объединения 3-х списков.
2. В списке найти разность между максимальным и минимальным элементами.

##### **Вариант 9.**

1. Дан список товаров мебельного магазина. Проверит, входит ли в список наименование «стол», если да, то исключить его.
2. Исключить из списка заданное число и найти длину оставшегося списка

##### **Вариант 10.**

1. В списке товаров продуктового магазина найти номер «молока», если он первый, то заменить «Молоко» на «Кефир».
2. Найти в списке минимальный элемент. Если это голова, то удалить все вхождения этого элемента. Если нет, то заменить минимальный элемент на число 999.

**Тестовые задания по курсу  
«Языки и методы программирования»**

Тестовые задания по курсу «Языки и методы программирования» разработаны для использования в процедурах промежуточного тестирования с целью оценки уровня подготовки студентов. Уровень сложности заданий и их содержание полностью соответствует требованиям ФГОС для педагогических специальностей. Условия проведения тестирования описаны в инструкции к тесту.

Назначение: Контроль знаний при промежуточном и итоговом тестировании  
 Время выполнения: 90 минут  
 Количество заданий: 30  
 Тип заданий: закрытый

*Форма тестовых заданий*

Тест состоит из заданий с выбором одного ответа из четырех предложенных. Ответы указываются на специальном бланке с таблицей номеров заданий.

*Алгоритм проверки*

- за правильный ответ испытуемый получает 1 балл,
- за неправильный или неуказанный ответ — 0 баллов.

№	Раздел учебной программы	Тема	Номера заданий
1	Входной модуль	Введение. История языков программирования. Классификация языков программирования. Основные алгоритмические конструкции	1, 2, 4
2	Модуль 1. Базовые конструкции языка C	Типы данных, их особенности. Операторы языка	7-13, 29, 30
		Ввод-вывод данных. Спецификаторы и модификаторы ввода-вывода	6, 14, 16, 17, 32
		Ветвления в языке C. Множественный выбор	18, 21, 22, 24, 27
		Циклические конструкции	3, 5, 19, 20, 23, 25, 26
		Работа с файлами	28
2	Модуль 2. Указатели и массивы в C	Понятие указателя. Операции с указателями	31
		Функции, передача данных в функцию. Формальные и фактические параметры	33-38, 42, 44, 45
		Статические одномерные массивы	43
		Статические двумерные массивы	106, 107
		Динамические одномерные и двумерные массивы	108, 109
3	Модуль 3. Структуры и списки в C	Структуры. Описание структур. Обращение к полям структуры	110
		Однонаправленные списки	39
		Стеки	39
		Очереди	40
		Деки	41
		Бинарные деревья	111, 112
4	Модуль 4. Основы параллельного программирования	Введение в параллельные вычисления. Общие понятия	46-50
		Библиотека MPI. Основные функции	51-54
		Парные операции передачи сообщений. Режимы передачи сообщений	55-63
		Широковещательная рассылка и редукция данных	64-75
		Разделение данных на части при передаче процессу	76-82
		Декомпозиция задачи	45, 46
5	Модуль 5. Основы объектно-ориентированного программирования	Классы и объекты	83
		Инкапсуляция	84, 85, 88
		Наследование	86, 89, 90
		Полиморфизм. Перегрузка операторов	87, 92, 93
		Шаблоны классов	91, 94
6	Модуль 6. Основы	Атомы и S-выражения	95, 96

	функционального программирования	Ветвления в Лиспе	97, 100
		Циклы и функции в Лиспе	98, 99
7	Модуль 7. Основы логического программирования	Структура программы в Прологе. Предикаты	101, 102, 103
		Рекурсия и обработка списков	104, 105

## Языки и методы программирования.

### Банк тестовых вопросов

1. Какой язык из перечисленных наиболее эффективен для программирования искусственного интеллекта?
  - a. LISP
  - b. FORTRAN
  - c. ALGOL
2. Какие языки относятся к языкам объектно-ориентированного программирования?
  - a. FORTRAN
  - b. Java
  - c. C++
  - d. LISP
3. Какой из циклов оптимальней использовать для повторения оператора(ов), если известно количество шагов цикла?
  - a. Цикл "while"
  - b. Цикл "repeat Until"
  - c. Цикл "for"
4. Какие языки относятся к алгоритмическим языкам?
  - a. LISP
  - b. PERL
  - c. Pascal
5. Какой из циклов нужно использовать, если известно, что тело цикла должен выполняться хотя бы один раз?
  - a. Цикл "while"
  - b. Цикл "repeat Until"
  - c. Цикл "for"
6. Для того чтобы вывести символ новой строки, надо:
  - a. закончить оператор точкой с запятой
  - b. воспользоваться специальным макросом endl
  - c. при выводе строки символов перевод строки добавляется автоматически
7. Компилятор языка Си++:
  - a. переводит текст программы в машинные инструкции
  - b. выполняет программу
  - c. форматирует текст программы так, чтобы его было удобно читать
8. Комментарий в программе на Си++
  - a. содержит указания компилятору по настройке программы
  - b. содержит пояснения к тексту и не оказывает влияния на выполнение программы
  - c. должен содержать допустимые аргументы программы
9. При выходе из функции main
  - a. программа повторяется с теми же аргументами
  - b. программа заканчивается
  - c. выполняется функция finish, определенная программистом
10. Объявление переменной
  - a. необходимо сделать до того, как использовать эту переменную
  - b. можно сделать в любой момент до завершения работы программы
  - c. желательно сделать для лучшего понимания программы, но можно и опустить
11. Выберите правильное объявление константы pi:
  - a. const float pi = 3.14;
  - b. float pi = (const) 3.14;
  - c. const float pi; pi = 3.14;
12. Отметьте правильное определение константы:
  - a. const long BITS = 32;
  - b. const bit ZERO = 0x0;
  - c. const float 0 Ora = 5.9787;
13. Укажите пункты, в которых происходит объявление констант и переменных

- a. float dD;  
 b. float sd2 = 3.2;  
 c. bType = 3;  
 d. int k; k = 89;
14. Укажите, в каких выражениях используются ключевые слова?  
 a. sdf = 2; int r = 24;  
 b. TStringList \*S = new TStringList;  
 c. x = 3; x = x + 4;  
 d. void function()
15. Если после выражения стоит точка с запятой, то  
 a. это оператор-выражение, действие которого заключается в вычислении выражения  
 b. выражение вычисляется, а его значение запоминается в специальной переменной, которую можно использовать в следующем операторе  
 c. выражение вычисляется только если первой стоит операция присваивания
16. Чему равен результат вычисления выражения  $x + 3 * b + x$  при  $x = 12$  и  $b = 8$  ?  
 a. 132  
 b. 48  
 c. 300
17. Каково будет значение переменной k после выполнения оператора  $k = ++ k$ ; если до его выполнения k равнялось 6?  
 a. 6  
 b. 7  
 c. 8
18. Если  $int i=3$ , какой будет результат вычисления:  
`if (i == 4) cout << "aaa";`  
`else if (i == 3) cout << "bbb";`  
`else if (i != 3) cout << "ccc";`  
 a. aaa  
 b. bbb  
 c. ccc  
 d. aaaccs  
 e. bbbccc  
 f. ошибка компиляции
19. Что выведет на экран следующая программа ?  

```
# include < iostream. h>
int main () {
int i;
for(i = 0; i < 9; i++)
cout << i+1;
return 1;
}
```

 a. цифры от 0 до 8  
 b. цифры от 1 до 9  
 c. программа не будет построена из-за ошибок
20. Укажите, какой будет результат вычисления k?  

```
int func()
{
int k = 10;
for (int i = 0; i <= k; i++)
{
return;
k = i; }
}
```

 a. 0  
 b. 10  
 c. 1  
 d. бесконечный цикл  
 e. 100
21. Если  $int n=45$ , какой будет результат?  
`switch(n) {`

```
case 23: cout << "aaa";  
case 45: cout << "bbb";  
default: cout << "vvv";  
break; }
```

- a. ошибка компилятора
- b. aaa
- c. bbb
- d. vvv
- e. bbbvvv
- f. aaavvv
- g. неопределенное поведение

22. При выполнении фрагмента кода

```
int x = 3, y = 2, z = 1;  
if(x >= y)  
if(y <= z)  
cout << "Вариант 1";  
else  
if( x >= z)  
cout << "Вариант 2";  
else  
cout <<"Вариант 3";
```

будет напечатано:

- a. Вариант 1
- b. Вариант 2
- c. Вариант 3
- d. ничего не будет напечатано
- e. программа не откомпилируется

23. Укажите, каким будет значение k в результате вычислений?

```
int func()  
{  
int k = 10;  
for (int i = 0; i <= k; i++)  
{  
return;  
k = i; }  
}
```

- a. 0
- b. 10
- c. 1
- d. бесконечный цикл
- e. 100

24. Если int n=3, какой будет результат?

```
switch(n) {  
case 2: cout << "aaa"; break;  
case 3: cout << "bbb"; break;  
default: cout << "vvv"; break; }
```

- a. ошибка компилятора
- b. aaa
- c. bbb
- d. vvv
- e. неопределенное поведение

25. Что произойдет при выполнении:

```
for ( int i = 0; i < 5; ) {  
continue; i--; func(); }
```

- a. функция func выполнится 5 раз
- b. функция func не выполнится ни разу
- c. функция func будет выполняться бесконечно

26. Укажите, какие операторы используются для циклического выполнения кода?

- a. if (...) then ... else ...
- b. while (...) { ... }

- c. `switch (...) {}`
- d. `for (...; ...; ...) {...}`

27. Укажите результат вычисления цикла:

```
int k = 0;
do
{
k++;
if (k == 1) continue;
else break;
++k;
} while (k < 5);
```

- a. `k = 2;`
- b. `k = 0;`
- c. `k = 5;`
- d. `k = 4;`

28. Какой из наборов перечисляемых значений записан правильно?

- a. `enum { a, b, 3, 4 };`
- b. `enum { a, b = 3, c = 4, d = 3 };`
- c. `enum { a, b = 3, c, d };`

29. Каким будет результат работы следующего фрагмента программы:

```
int a = 5^3;
float b = 1.5f;
b += -- a/2;
cout << b;
```

- a. 63.50
- b. 64.00
- c. 3.50
- d. 4.00
- e. 63.00

30. Найдите недопустимую запись символьной константы:

- a. `' F'`
- b. `"\022"`
- c. `'Ю'`
- d. `\ a'`

31. Какой результат вычисления следующего выражения?

```
0xFF & 5 >> 1 + 1
```

- a. 1
- b. 2
- c. 256

32. Каким будет результат следующей программы:

```
int a = 5*3;
float b = 1.5f;
b += -- a/2;
cout << b;
```

- a. 8.5
- b. 9.0
- c. 8.0
- d. 9.5
- e. 7.5

33. Переменные, создаваемые в теле функции, называются локальными, потому что:

- a. они существуют только во время выполнения функции
- b. к ним можно обращаться только после их объявления
- c. их имена должны начинаться с символа l

34. Отметьте допустимые имена функций:

- a. `_This_Function_12_x_`
- b. `ax%u7`
- c. `ffffffAAAAAA`
- d. `3_pi`
- e. `calculateIt`
- f. `thisname_is_too_long_forafunction`

35. Как называется функция, которая вызывает саму себя?

- a. конструктор
- b. деструктор
- c. подставляемая
- d. рекурсивная

36. В чем разница между фактическими и формальными параметрами?

- a. формальные параметры могут использоваться только вне тела функции, а фактические - используются как вне функции, так и внутри её
- b. формальные параметры определены в теле функции, а фактические - значения, с которыми функция вызывается
- c. нет различий

37. Укажите правильное объявление функции

- a. `int MyFunc(double x, y, int i)`
- b. `void correct(double d = 3.14, double a)`
- c. `int sum(int j, int k, int x = 0)`

38. Какое из следующих утверждений об операторе return является верным?

- a. оператор return должен стоять последним в теле функции
- b. оператор return завершает выполнение функции
- c. в теле функции должен присутствовать только один оператор return

39. Имеется следующее описание:

Type

```
U = ^Zveno;  
Zveno = Record  
    X, Y : Boolean;  
    Pred, Next : U
```

End;

Var

```
Logic : Boolean;  
A, B : Pointer;  
X, Y : U;
```

К ошибке компиляции "Несовместимость типов" приведет следующее присваивание:

- a. `A := X^.Next^.Next`
- b. `X := Y`
- c. `Logic := X^.X`
- d. `X^.Next := A`
- e. `X^ := Y^.Next`

40. Список объявлен следующим образом:

Type

```
Ukaz = ^Zveno;  
Zveno = Record  
    X : String;  
    N : Ukaz
```

End;

Var

```
First : Ukaz; {ссылка на начало списка}
```

В следующем фрагменте программы:

```
P := First;
```

```
While P^.N <> Nil Do
```

```
Begin
```

```
  B := P; M := P;
```

```
  While B <> Nil Do
```

```
    Begin
```

```
      If B^.X < M^.X Then
```

```
        M := B; B := B^.N
```

```
    End;
```

```
  S := P^.X;
```

```
  P^.X := M^.X;
```

```
  M^.X := S;
```

```
  P := P^.N
```

```
End;
```

выполняется

- a. перемещение компонента к началу списка;
- b. сортировка компонентов списка в порядке возрастания;
- c. сортировка компонентов списка в порядке убывания;
- d. перестановка соседних компонентов списка;
- e. добавление в список нескольких новых компонент.

41. Список объявлен следующим образом:

Type

SS = ^List;

List = Record

A : LongInt;

Next : SS

End;

В приведённом фрагменте программы (First — ссылка на первый элемент списка)

P := First;

S := 0;

While Not (P = Nil) Do

Begin

S := S + 1;

P := P^.Next

End;

определяется

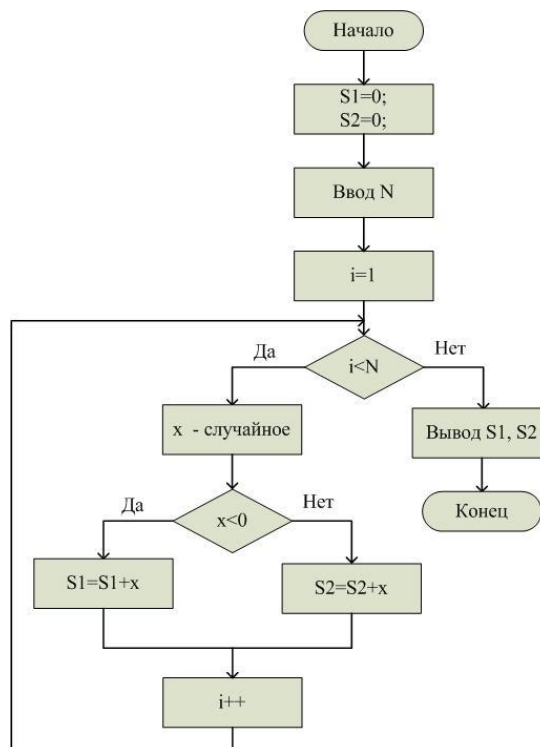
- a. первый элемент списка;
- b. сумма элементов списка;
- c. сумма первого и последнего элементов списка;
- d. количество элементов списка;
- e. количество звеньев списка, где указатель на следующее звено не Nil.

42. Даны натуральные числа a, b, обозначающие соответственно числитель и знаменатель дроби. Сократить дробь, т.е. получить правильную несократимую дробь. Какую часть в предложенной задаче вы можете выделить как функцию?

43. Даны две дроби A/B и C/D (A, B, C, D — натуральные числа). Составить программу для деления дроби на дробь. Результат должен быть несократимой дробью. Какую часть в предложенной задаче вы можете выделить как подпрограмму?

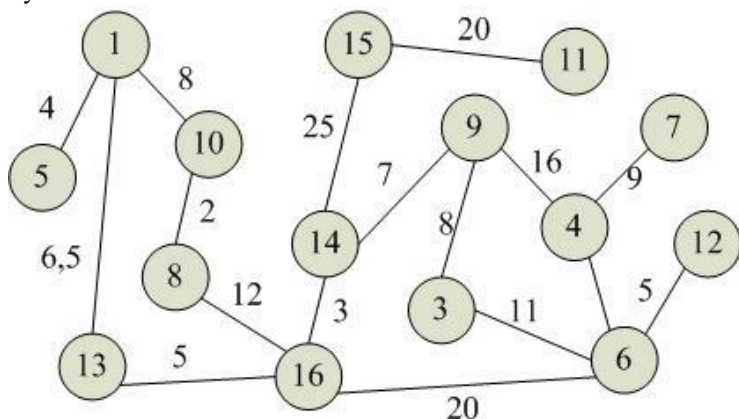
44. Проанализировав схему, предположите, решение какой задачи отображено.

45. Проанализировав схему, предположите, решение какой задачи отображено.





46. На рисунке изображены дороги, соединяющие населённые пункты. Населённые пункты пронумерованы, цифра возле дороги означает её протяжённость. Разделите схему на районы. Каждый район должен обслуживаться одним автобусом. Между каждым из районов курсирует один автобус. Ответ представьте в виде: номер района1 [перечень номеров населённых пунктов]; номер района2 [перечень номеров населённых пунктов] и т.д.; автобусов нужно - N.



47. Отметьте, какие утверждения о массивно-параллельных компьютерах верны?  
 a) массивно-параллельные компьютеры не могут работать без хост-машины;  
 b) основными отличительными характеристиками массивно-параллельных компьютеров являются: используемые микропроцессоры и коммуникационная сеть (среда);  
 c) в массивно-параллельных компьютерах не может быть больше 1024 процессоров;  
 d) коммуникационная сеть массивно-параллельных компьютеров всегда организуется в виде трёхмерного тора.

48. Отметьте верные утверждения о кластерных вычислительных системах:  
 a) кластерные системы строятся на базе серийных процессоров;  
 b) на каждом узле кластера исполняется свой экземпляр операционной системы;  
 c) максимально возможное число процессоров кластерной системы равно 128;  
 d) один кластер может строиться с использованием нескольких коммуникационных технологий.

49. Разновидностью распараллеливания являются технологии и приёмы:

- a) суперскалярности;
- b) структурного программирования;
- c) многопроцессорности;
- d) объектно-ориентированного программирования.

50. Отметьте верные утверждения:

- a) MPI - это сокращение от Message Passing Interface.
- b) Функция MPI\_Comm\_size определяет общее число запущенных параллельных процессов приложения.
- c) Каждый параллельный процесс в MPI имеет номер.
- d) MPI - это сокращение от Multiple Parallel Interface.

51. Какая операция MPI не относится к коллективным операциям?

- a) MPI\_Bcast;
- b) MPI\_Send;
- c) MPI\_Reduce;
- d) MPI\_Gather.

52. Отметьте верные утверждения:

- a) использование функций MPI\_Send и MPI\_Recv может привести к тупиковой ситуации (deadlock);
- b) в коллективных операциях участвуют все процессы приложения;
- c) функция, соответствующая коллективной операции, должна быть вызвана каждым процессом, быть может, со своим набором параметров;
- d) возврат процесса из функции, реализующей коллективную операцию, всегда означает, что операция уже завершена;

53. Фрагмент программы:

```
MPI_Comm_size( comm, &size);
MPI_Send( buf, 15, MPI_INT, size+1, 8, comm);
```

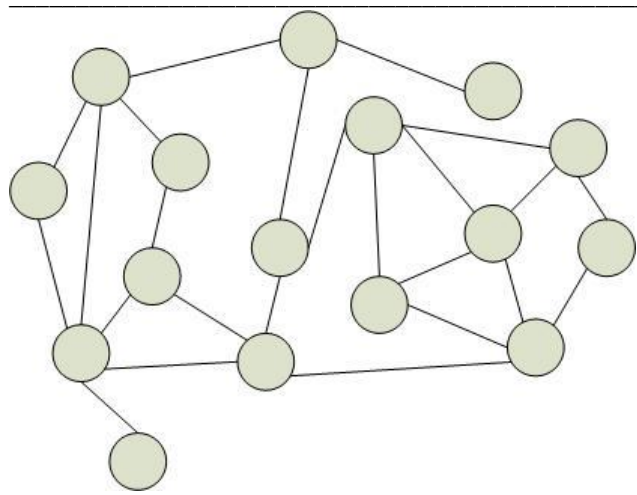
MPI\_Send( buf, 8, MPI\_INT, size-1, 15, comm);

- a) может быть правильным;
  - b) заведомо содержит одну ошибку;
  - c) заведомо содержит две ошибки;
  - d) заведомо содержит три ошибки.
54. Отметьте верные утверждения о функции MPI\_Recv:
- a) перед вызовом функции MPI\_Recv надо обратиться к функции MPI\_Get\_count;
  - b) возврат из функции означает, что либо произошла ошибка, либо принятое сообщение расположено в первом параметре;
  - c) функцией MPI\_Recv нельзя принимать сообщение, посланное с помощью функции MPI\_Ssend;
  - d) нельзя использовать функцию MPI\_Recv, если неизвестны отправитель сообщения или тег сообщения.
55. Псылка сообщения с блокировкой (MPI\_Send) означает, что возврат из функции произойдёт тогда, когда:
- a) можно повторно использовать параметры данной функции;
  - b) сообщение покинет процесс;
  - c) сообщение принято адресатом;
  - d) адресат инициировал приём данного сообщения.
56. Функция MPI\_Send возвращает:
- a) число байт, заявленное в вызове для пересылки;
  - b) значение MPI\_SUCCESS или код ошибки;
  - c) номер процесса, которому адресована передача;
  - d) число реально переданных байт.
57. В декартовой топологии множество процессов представляется в виде:
- a) прямоугольной решётки;
  - b) графа произвольного вида;
  - c) полного графа;
  - d) звезды.
58. MPI поддерживает топологии вида:
- a) прямоугольная решётка произвольной размерности и граф произвольного вида;
  - b) только прямоугольная решётка произвольной размерности;
  - c) только граф произвольного вида;
  - d) тор произвольной размерности и граф произвольного вида.
59. Операция широковещательной рассылки данных это:
- a) операция рассылки значений ведущим процессом всем остальным процессам коммутатора; все процессы получают часть исходных данных;
  - b) операция рассылки значений ведущим процессом всем остальным процессам коммутатора, все процессы получают рассылаемые данные целиком;
  - c) операция рассылки различающихся значений ведущим процессом всем остальным процессам коммутатора;
  - d) операция рассылки значений ведущим процессом всем остальным процессам приложения, все процессы получают рассылаемые данные целиком.
60. Операцию редукции данных MPI\_Reduce можно описать:
- a) операцией передачи данных, при которой над собираемыми значениями осуществляется та или иная обработка, при этом результат обработки получают все процессы коммутатора;
  - b) как операцию передачи данных, при которой над собираемыми значениями осуществляется обработка, при этом частичные значения результатов редуцирования получают все процессы параллельной программы;
  - c) как операцию передачи данных, при которой над собираемыми значениями осуществляется обработка в процессе передачи, при этом результат обработки получает только ведущий процесс;
  - d) операцией передачи данных, при которой все процесса коммутатора получают различающиеся значения.
61. В коллективных операциях передачи данных обязаны принимать участие:
- a) все процессы программы;
  - b) все процессы группы коммутаторов;
  - c) все процессы одного коммутатора;
  - d) некоторые процессы одного коммутатора.
62. Режим передачи по готовности может быть использован только если:
- a) операция приёма сообщения уже инициирована;

- b) операция приёма сообщения гарантированно будет запущена позднее момента начала передачи сообщения;
- c) при размере сообщения, меньшем размера системного буфера.
63. В буферизованном режиме функция отправки сообщения завершается:
- a) сразу же после копирования сообщения в системный буфер;
- b) при получении от процесса-получателя подтверждения о начале приёма отправленного сообщения;
- c) при начале фактической передачи сообщения.
64. В синхронном режиме передачи завершение функции отправки сообщения происходит:
- a) при старте передачи данных процессом-отправителем по сети;
- b) при завершении копирования сообщения в системный буфер;
- c) при получении от процесса-получателя подтверждения о начале приёма отправленного сообщения, при этом отправленное сообщение или полностью принято процессом-получателем или находится в состоянии приёма.
65. Приём сообщений при помощи функции MPI\_Recv может быть выполнен:
- a) от любого адресата и с любым тегом при указании специальных значений в качестве параметров вызова функции,
- b) от любого адресата, однако, тег сообщения должен быть указан однозначно,
- c) только от однозначно определяемого адресата с заданным тегом.
66. Прием сообщения при помощи функции MPI\_Recv может быть инициирован:
- a) только до момента начала отправки сообщения;
- b) только после момента начала отправки сообщения;
- c) до момента, в момент или после момента начала отправки сообщения;
- d) только в момент начала отправки сообщения.
67. Все данные для передачи в качестве сообщения MPI описываются с помощью триады:
- a) адрес памяти, ранг процесса-отправителя, используемый коммуникатор,
- b) адрес памяти, ранг процесса-получателя, используемый коммуникатор.
- c) адрес памяти, количество и тип элементов данных,
68. Функция MPI\_Recv:
- a) в зависимости от используемой операции передачи может как заблокировать, так и не заблокировать процесс-получатель.
- b) блокирует процесс-получатель до момента фактического получения сообщения,
- c) принимает сообщение в фоновом режиме, процесс в это время может продолжать вычисления.
69. Указание используемого коммуникатора является:
- a) обязательным для всех операций передачи данных в MPI,
- b) необязательным для некоторых операций передачи данных в MPI,
- c) обязательным для некоторых операций передачи данных в MPI.
70. Под коммуникатором в MPI понимается:
- a) группа процессов, в рамках которой выполняются операции передачи данных,
- b) пара процессов, в рамках которой происходит информационное взаимодействие.
- c) специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров, используемых при выполнении операций передачи данных,
71. Под параллельной программой в рамках MPI понимается:
- a) множество одновременно работающих процессоров.
- b) множество одновременно выполняемых процессов,
- c) множество одновременно выполняемых потоков,
72. Процессы параллельной программы в рамках MPI:
- a) могут выполняться на разных процессорах, на одном процессоре могут располагаться несколько процессов,
- b) обязательно выполняются на одном процессоре.
- c) могут выполняться только на разных процессорах,
73. Завершение вызова функции неблокирующего обмена означает:
- a) фактическое выполнение обмена данными;
- b) фактическое выполнение приема данных (для функции неблокирующего приема) или начало фоновой передачи (для функции неблокирующей передачи);
- c) инициацию запрошенной операции передачи, но ничего не говорит о начале или завершённости обмена.

74. Дана матрица размером  $N \times M$ . Написать программу транспонирования матрицы. Предложите способ разбиения задачи на подзадачи, опишите информационные связи между подзадачами.

75. На рисунке изображён граф информационных зависимостей подзадач некоторой задачи. Каким образом можно укрупнить подзадачи? Каким будет граф после укрупнения?



76. Даны матрицы размером  $N \times M$  и  $M \times N$ . Написать программу умножения матриц. Предложите способ разбиения задачи на подзадачи, опишите информационные связи между подзадачами.

77. Как передать с 1-го процесса на 8-й целое число 5 в рамках коммуникатора `MPI_COMM_WORLD`?

78. Как передать с 3-го процесса на 0-й массив вещественных чисел из 10 элементов в рамках коммуникатора `MPI_COMM_WORLD`?

79. Как передать с 0-го процесса всем остальным массив из 20 символов в рамках коммуникатора `MPI_COMM_WORLD`?

80. На нулевом процессе сформирован массив из 100 целых чисел. Как переслать равные части массива на все процессы коммуникатора `MPI_COMM_WORLD`, если в коммуникатор входит 10 процессов?

81. На 10 процессах коммуникатора `MPI_COMM_WORLD` вычислены промежуточные значения. Напишите функцию, которая перемножит данные с 10-ти процессов и передаст результат на 0-й процесс.

82. На 15 процессах коммуникатора `MPI_COMM_WORLD` вычислены промежуточные значения. Напишите функцию, которая найдёт максимальное значение и передаст результат на 0-й процесс.

Даны два вектора. Нужно найти их скалярное произведение. Предложите способ разбиения данной задачи на подзадачи, которые могут выполняться параллельно. Опишите характер связей между предложенными подзадачами.

83. Для обращения к полям и методам класса используется оператор:

- а) присваивания;
- б) "звездочка";
- в) логического сравнения;
- г) «точка»

84. Допишите конструктор в класс `Time`, который принимает время в секундах.

```
class Time {
    int hour, min, sec;
    // конструктор расположить здесь
    // ...
public:
    string toString() {
        return hour + ":" + min + ":" + sec;
    }
}
```

Так, чтобы при создании и печати объекта выводилось 1:1:1 (один час, одна минута, одна секунда)

85. Какой модификатор доступа может быть у полей и методов класса, чтобы они были видны только внутри этого класса.

86. Имеется класс Person.

```
class Person{
public:
    string name;
    int age;
}
```

Опишите его подкласс Employee таким образом, чтобы в объекте этого типа были доступны следующие публичные элементы:

- строковые переменные name(имя) и company(компания) и целочисленная переменная age(возраст).
- методы work(работать) и relax(отдыхать), оба без параметров и возвращаемого значения.

87. Создайте класс Dog, наследник Animal, и реализуйте в нем полиморфный метод eat(), возвращающий строку "Bone".

```
class Animal{
public:
    string eat();
}
```

88. Выберите правильные утверждения

- конструкторы класса не наследуются
- конструкторов класса может быть несколько, их синтаксис определяется программистом
- конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- конструктор возвращает указатель на объект
- конструктор не возвращает значение

89. Выберите правильное объявление производного класса

- class MoreDetails:: Details;
- class MoreDetails: public class Details;
- class MoreDetails: public Details;
- class MoreDetails: class(Details);

90. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового

- ключ доступа - public; в базовом классе: private; права доступа в производном классе - protected
- ключ доступа - любой; в базовом классе: private; права доступа в производном классе - нет прав
- ключ доступа - protected или public ; в базовом классе: protected; права доступа в производном классе - protected
- ключ доступа - private; в базовом классе: public; права доступа в производном классе - public
- ключ доступа - любой; в базовом классе: public; права доступа в производном классе - такие же, как ключ доступа

91. Шаблон функции - это...

- определение функции, в которой типу обрабатываемых данных присвоено условное обозначение
- прототип функции, в котором вместо имен параметров указан условный тип
- определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
- определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

92. Полиморфизм – это :

- средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
- средство, позволяющее в одном классе использовать методы с одинаковыми именами;
- средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
- средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

93. Полиморфизм реализован через механизмы:

- перегрузки функций, виртуальных функций, шаблонов
- перегрузки функций, наследования методов, шаблонов;
- наследования методов, виртуальных функций, шаблонов
- перегрузки функций, наследования, виртуальных функций.

94. Отметьте верные утверждения

- а) можно явным образом указывать параметры шаблонов в угловых скобках;
- б) для функций могут быть заданы параметры шаблона по умолчанию, используя знак = после объявления параметра шаблона
- в) шаблонные параметры, указанные явным образом должны совпадать с типами параметров, переданных в функцию
- г) шаблонные функции можно перегружать

95. Какой результат будет получен в результате вычисления выражения

```
(LET ((i 0) (j 5))
  (DOTIMES (k j)
    (SETQ i (+ i k))
  )
)
```

- a) 5
- б) T
- в) NIL
- г) 15
- д) 10

96. Что будет выведено?

```
(car (= 2 3))
```

- a) nil
- б) =
- в) ‘
- г) quote
- д) ‘(= 2 3)
- е) ошибка

97. Какой результат вернет выражение (ATOM ‘(N M))?

- a) (N M)
- б) nil
- в) T
- г) N
- д) N M

98. Какое значение вернет выражение (EXPT 2 3)?

- a) nil
- б) 6
- в) 8
- г) T
- д) 9

99. Укажите результат команды (CDADR’(a (b (c) d) e))

- a) (c)
- б) e
- в) ()
- г) (b (c) d)
- д) ((c) d)

100. Что из перечисленного относится к циклам в Lisp?

- a) loop
- б) dotimes
- в) do
- г) dolist
- д) while

101. Логическая модель знаний состоит из ...

- a) фактов и правил;
- б) фактов;
- в) правил;
- г) предложений;
- д) заявлений.

102. Что такое факт?

- a) это логическая модель знаний;
- б) это утверждение общего характера;
- в) это утверждение правила;
- г) это частное утверждение;
- д) нет правильного ответа.

103. Запись сын(A,B):-отец(B,A) означает:

- a) ЕСЛИ B – отец A, ТО A является сыном B;
- б) ЕСЛИ A – отец B, ТО B является сыном A;
- в) ЕСЛИ A – сын B, ТО B является отцом A;
- г) ЕСЛИ B – сын A, ТО A является отцом B;
- д) нет правильного ответа.

104. Упорядоченный набор объектов одного и того же типа называется:

- А) массивом;
- Б) таблицей;
- В) списком;
- Г) множеством.

105. Операция отделения «головы» обозначается:

- А) (Head Tail);
- Б) [Head / Tail];
- В) (Head | Tail);
- Г) [Head | Tail].

106. Пусть  $\text{int } i=16, *p, \text{ar}[]=\{0,1,2,3,4\};$

Чему будет равно значение переменной  $i$  после выполнения оператора:

```
for (p=ar+4; p>=ar; p--) i=*p;?
```

107. Элементы массива  $\text{int } p[5]$  равны соответственно 1, -1, 5, 2, 4. Значение выражения

$p[0] * p[2] - p[2] * p[1] + p[p[4] - p[0]]$

равно: \_\_\_\_\_

108. В каком из вариантов ответов объявлен двумерный массив?

- а) `int array[20, 20];`
- б) `array anarray[20][20];`
- в) `char array[20];`
- г) `int anarray[20][20];`

109. Какое значение будет напечатано, в результате выполнения следующего кода?

```
#include <iostream>
int main()
{
    int sum = 0;
    int array[3][] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};
    for (int i = 0; i < 3 ; ++i)
    {
        for (int j = 2; j < 3 ; j++)
        {
            sum += array[i][j];
        }
    }
    std::cout << sum << std::endl;
    return 0;
}
```

- а) 9
- б) 15
- в) 21
- г) синтаксическая ошибка

110. В данном коде представлен доступ к участнику структуры.

```
int i1 = v.sz;
```

Выберите варианты описания доступа, которые могут соответствовать этому коду

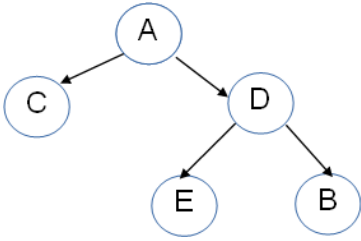
- а) Доступ через имя
- б) Доступ через ссылку
- в) Доступ через указатель

111. Программа получает на вход последовательность целых чисел и строит из них двоичное дерево поиска. Балансировка дерева не производится.

В ответе укажите единственное число - высоту получившегося дерева для следующей последовательности чисел, считая, что дерево из одного элемента имеет высоту 1:

2 3 5 6 8

112. Какие значения могут принимать А, В, С, D, Е, чтобы дерево было деревом двоичного поиска.



Выберите один ответ:

A=1, B=2, C=3, D=4, E=5

A=2, B=3, C=1, D=4, E=5

A=2, B=5, C=1, D=4, E=3

A=2, B=5, C=1, D=3, E=4



### **3.3. Анализ результатов обучения и перечень корректирующих мероприятий по учебной дисциплине**

#### **Лист внесения изменений**

Дополнения и изменения в учебной программе на 2017/2018 учебный год

1. Список литературы обновлен учебными и учебно-методическими изданиями, электронными образовательными ресурсами.
2. Обновлен перечень лицензионного программного обеспечения.

Рабочая программа пересмотрена и одобрена на заседании кафедры 03 мая 2017 г. протокол № 10

Рабочая программа пересмотрена и одобрена на заседании НМСС 26 мая 2017 г. протокол № 9

Внесенные изменения утверждает



Заведующий кафедрой \_\_\_\_\_ Пак Н.И.

## Лист внесения изменений

Дополнения и изменения в рабочей программе дисциплины на 2018/2019 учебный год:

В рабочую программу дисциплины вносятся следующие изменения:

1. Список литературы обновлен учебными и учебно-методическими изданиями, электронными образовательными ресурсами. Обновлен перечень современных профессиональных баз данных и информационных справочных систем.
2. В фонд оценочных средств внесены изменения в соответствии с приказом «Об утверждении Положения о фонде оценочных средств для текущего контроля успеваемости, промежуточной и итоговой (государственной итоговой) аттестации» от 28.04.2018 № 297 (п)
3. На титульном листе РПД и ФОС изменено название ведомственной принадлежности «Министерство науки и высшего образования» на основании приказа «о внесении изменений в сведения о КГПУ им. В.П. Астафьева» от 15.07.2018 № 457 (п).

Рабочая программа пересмотрена и одобрена на заседании кафедры 04 апреля 2018 г. протокол № 7

Внесенные изменения утверждаю:



Заведующий кафедрой \_\_\_\_\_ Пак Н.И.

Рабочая программа пересмотрена и одобрена на заседании НМСС 23 мая 2018 г. протокол № 8 Внесенные изменения утверждаю.



Председатель \_\_\_\_\_ Бортновский С.В.

### Лист внесения изменений

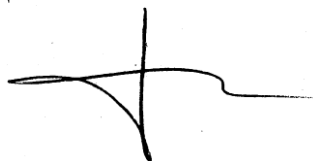
Дополнения и изменения рабочей программы на 2019/2020 учебный год

В рабочую программу вносятся следующие изменения:

- На титульном листе РПД и ФОС изменено название ведомственной принадлежности «Министерство науки и высшего образования» на основании приказа «о внесении изменений в сведения о КГПУ им. В.П. Астафьева» от 15.07.2018 № 457 (п).

Рабочая программа пересмотрена и одобрена на заседании кафедры 08 мая 2019 г. протокол № 9

Внесенные изменения утверждаю:



Заведующий кафедрой \_\_\_\_\_

Пак Н.И.

Одобрено научно-методическим советом ИМФИ

16 мая 2019 г. протокол № 8



Председатель \_\_\_\_\_

Бортновский С.В.



#### 4. УЧЕБНЫЕ РЕСУРСЫ

##### 4.1.КАРТА ЛИТЕРАТУРНОГО ОБЕСПЕЧЕНИЯ ДИСЦИПЛИНЫ

ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

для обучающихся образовательной программы

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность

(профиль) образовательной программы

Математика и информатика

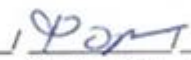
по очной форме обучения

Наименование	Место хранения/ электронный адрес	Кол-во экземпляров/точек доступа
<b>ОСНОВНАЯ ЛИТЕРАТУРА</b>		
Шилдт Герберт С++. Полное руководство/ Г.Шилдт. – 4-е издание, - Вильямс, 2017. - 800 с.: ил. - ISBN: 978-5-8459-2140-6,	ОБИМФИ	9
Страуструп Бьерн. Язык программирования С++/ Б.Страуструп. – Бином, 2018. – 1136 с.: ил. – ISBN: 978-5-9518-0425-9,	ОБИМФИ	9
Васильев А.Н. Программирование на С в примерах и задачах. / А.Н.Васильев – Эксмо-пресс, 2017, - 368с., ISBN: 978-5-699-87445-3	ОБИМФИ	9
Богачев К.Ю.: Основы параллельного программирования / К.Ю.Богачев – Бином: Лаборатория знаний, 2018г., - 342 с, - ISBN: 978-5-9963-1616-8	ОБИМФИ	9
Гергель В.П.: Теория и практика параллельных вычислений. Учебное пособие / В.П.Гергель - Бином: Лаборатория знаний, 2018г. – 423 с. - ISBN: 978-5-94774-645-7	ОБИМФИ	9
Сайбель Питер Практическое использование Common Lisp / П.Сайбель – ДМК-пресс, 2017 – 488 с. - ISBN: 978-5-94074-627-0	ОБИМФИ	9
Ездаков А. Функциональное и логическое программирование. Учебное пособие / А. Ездаков - Бином: Лаборатория знаний, 2017, - 119 с. - ISBN: 978-5-9963-0575-9	ОБИМФИ	9
Марков В.Н. Современное логическое программирование на языке Visual Prolog 7.5 / В.Н. Марков – BHV, 2016 – 544с. - ISBN: 978-5-9775-3487-1	ОБИМФИ	9
<b>ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА</b>		
Кнут Дональд Искусство программирования, том 1/ Д.Кнут. – Вильямс, 2019. – 720 с. – ISBN: 978-5-8459-1984-7	ОБИМФИ(9)	9
Кнут Дональд Искусство программирования, том 2/ Д.Кнут. – Вильямс, 2019. – 832 с. – ISBN: 978-5-8459-0081-4	ОБИМФИ(9)	9
Кнут Дональд Искусство программирования, том 3/ Д.Кнут. – Вильямс, 2019. – 832 с. – ISBN: 978-5-907144-41-5	ОБИМФИ(9)	9

Макконелл Стив. Совершенный код. Мастер-класс / С.Макконелл. – Русская редакция, 2017, 896с. - ISBN: 978-5-7502-0064-1	ОБИМФИ(9)	9
Борзунов С.В., Кургалин С.Д., Флегель А.В. Практикум по параллельному программированию / Борзунов С.В., Кургалин С.Д., Флегель А.В. – ВНУ, 2017, - 236 с. - ISBN: 978-5-9909805-0-1	ОБИМФИ(9)	9
<b>УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ</b>		
Электронный учебный курс «Языки и методы программирования» авт. Сокольская М.А., КГПУ им. В.П.Астафьева URL: <a href="http://e.kspu.ru/course/view.php?id=117">http://e.kspu.ru/course/view.php?id=117</a>	Электронный университет сайт КГПУ им. В.П. Астафьева	Индивидуальный доступ
<b>РЕСУРСЫ СЕТИ ИНТЕРНЕТ</b>		
Программирование на языке C++	<a href="https://stepik.org/course/7">https://stepik.org/course/7</a>	Свободный доступ
Параллельное программирование (на английском языке)	<a href="https://www.coursera.org/learn/parprog1">https://www.coursera.org/learn/parprog1</a>	Свободный доступ
Суперкомпьютеры и параллельная обработка данных	<a href="https://openedu.ru/course/msu/PARP/ROG/">https://openedu.ru/course/msu/PARP/ROG/</a>	Свободный доступ
Теория алгоритмов	<a href="http://portal.tpu.ru:7777/SHARED/s/SHEFER/Study/Tab2/Tab3/Razdel_N6.pdf">http://portal.tpu.ru:7777/SHARED/s/SHEFER/Study/Tab2/Tab3/Razdel_N6.pdf</a>	Свободный доступ
Функциональное программирование: базовый курс	<a href="https://openedu.ru/course/ITMOUniversity/FPBC/">https://openedu.ru/course/ITMOUniversity/FPBC/</a>	Свободный доступ
<b>ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ И ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ</b>		
Научная библиотека КГПУ им. В.П. Астафьева	<a href="http://library.kspu.ru/jirbis2/">http://library.kspu.ru/jirbis2/</a>	Локальная сеть вуза
Межвузовская электронная библиотека (МЭБ)	<a href="https://icdlib.nspu.ru/">https://icdlib.nspu.ru/</a>	Индивидуальный неограниченный доступ
<a href="http://elibrary.ru">Elibrary.ru</a> [Электронный ресурс]: электронная библиотечная система : база данных содержит сведения об отечественных книгах и периодических изданиях по информатике / Рос. информ. портал. - Москва. 2000- . - Режим доступа: <a href="http://elibrary.ru">http://elibrary.ru</a> .	<a href="http://elibrary.ru">http://elibrary.ru</a>	Свободный доступ.
East View: универсальные базы данных [Электронный ресурс] : периодика России, Украины и стран СНГ , - Элктрон.дан. - ООО ИВИС. - 2011 - .	<a href="https://dlib.eastvirw.com/">https://dlib.eastvirw.com/</a>	Индивидуальный неограниченный доступ

Согласовано:

Главный библиотекарь  
(должность структурного подразделения)

 Фортова А.А.  
(подпись) (Фамилия И.О.)

## 4.2. Карта материально-технической базы дисциплины

ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

### для обучающихся образовательной программы

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность (профиль) образовательной программы Математика и информатика

по очной форме обучения

г. Красноярск, ул. Перенсона 7

(Корпус №4)

<b>Аудитория</b>	<b>Оборудование</b> <b>(наглядные пособия, макеты, модели, лабораторное оборудование, компьютеры, интерактивные доски, проекторы, программное обеспечение)</b>
	для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля успеваемости и промежуточной аттестации
Перенсона,7 <b>(Корпус №4)</b> № 4-02	<b>для проведения занятий лекционного типа</b> <b>Оборудование</b> Маркерная доска – 1 шт, ноутбук с выходом в интернет – 10 шт, мультимедийный демонстрационный комплекс (проектор, интерактивная доска, колонки, USB-камера) – 1 шт, система видеоконференцсвязи Policom – 1 шт, <b>Программное обеспечение</b> Альт Линукс Школьный - (Свободная лицензия) Microsoft® Windows® 7 Professional Лицензия Dreamspark (MSDN AA); Kaspersky Endpoint Security – Лиц сертификат №2304- 180417-031116- 577-384; 7-Zip - (Свободная лицензия GPL); Adobe Acrobat Reader – (Свободная лицензия); Google Chrome – (Свободная лицензия); Mozilla Firefox – (Свободная лицензия); LibreOffice – (Свободная лицензия GPL); XnView – (Свободная лицензия); Java – (Свободная лицензия);

	<p>DEV C++ (Свободная лицензия)  Lispbox (Свободная лицензия)  Putty (Свободная лицензия)  WinSCP (Свободная лицензия)  VLC – (Свободная лицензия).  Консультант Плюс - (Свободная лицензия для учебных целей);</p>
<p>Персона,7  (Корпус №4)  № 4-12</p>	<p style="text-align: center;"><b>для проведения лабораторных работ</b></p> <p><b>Оборудование</b>  Маркерная доска – 1 шт, ноутбук с выходом в интернет – 10 шт, мультимедийный демонстрационный комплекс (проектор, интерактивная доска, колонки, USB-камера) – 1 шт, система видеоконференцсвязи Policom – 1 шт,  <b>Программное обеспечение</b>  Альт Линукс Школьный - (Свободная лицензия)  Microsoft® Windows® 7 Professional Лицензия Dreamspark (MSDN AA);  Kaspersky Endpoint Security – Лиц сертификат №2304- 180417-031116- 577-384;  7-Zip - (Свободная лицензия GPL);  Adobe Acrobat Reader – (Свободная лицензия);  Google Chrome – (Свободная лицензия);  Mozilla Firefox – (Свободная лицензия);  LibreOffice – (Свободная лицензия GPL);  XnView – (Свободная лицензия);  Java – (Свободная лицензия);  VLC – (Свободная лицензия).  DEV C++ (Свободная лицензия)  Visual Prolog 9 (ограниченная, неисключительная лицензия только для некоммерческого и образовательного использования)  Lispbox (Свободная лицензия)  Putty (Свободная лицензия)  WinSCP (Свободная лицензия)  Консультант Плюс - (Свободная лицензия для учебных целей);</p>
	<b>для самостоятельной работы</b>
<p>Персона,7  (Корпус №4)  №1-02</p>	<p><b>Оборудование</b>  Компьютер-10 шт, принтер-1шт  <b>Программное обеспечение</b>  Microsoft® Windows® 7 Professional Лицензия Dreamspark (MSDN AA);</p>



<p>Kaspersky Endpoint Security – Лиц сертификат №2304- 180417-031116- 577-384; 7-Zip - (Свободная лицензия GPL); Adobe Acrobat Reader – (Свободная лицензия); Google Chrome – (Свободная лицензия); Mozilla Firefox – (Свободная лицензия); LibreOffice – (Свободная лицензия GPL); XnView – (Свободная лицензия); Java – (Свободная лицензия); DEV C++ (Свободная лицензия) Lispbox (Свободная лицензия) Visual Prolog 9 (ограниченная, неисключительная лицензия только для некоммерческого и образовательного использования) Putty (Свободная лицензия) WinSCP (Свободная лицензия) VLC – (Свободная лицензия). Консультант Плюс - (Свободная лицензия для учебных целей);</p>
---