

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  
**ВЫСШЕГО ОБРАЗОВАНИЯ**  
**КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ**  
**им. В.П. Астафьева**  
**(КГПУ им. В.П. Астафьева)**

**Институт математики, физики и информатики**  
**Базовая кафедра информатики и информационных технологий в**  
**образовании**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ**

**Направление: 44.03.05 Педагогическое образование**

**Профиль «Физика и информатика»**

Квалификация (степень): бакалавр

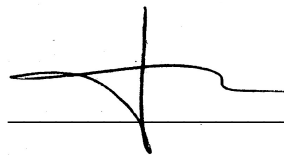
Очная форма обучения

Красноярск 2016

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

Рабочая программа дисциплины обсуждена на заседании кафедры ИИТвО протокол № 3 от 05.10.2016 г.

Заведующий кафедрой  
(ф.и.о., подпись)



Пак Н.И.

Одобрено научно-методическим советом ИМФИ  
26.10.2016

Председатель  
(ф.и.о., подпись)



Бортновский С.В.

## Содержание

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	4
ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ.....	7
ЛИСТ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ С ДРУГИМИ ДИСЦИПЛИНАМИ НАПРАВЛЕНИЯ И ООП.....	10
ТЕХНОЛОГИЧЕСКАЯ КАРТА ОБУЧЕНИЯ ДИСЦИПЛИНЕ.....	10
ТЕХНОЛОГИЧЕСКАЯ КАРТА УЧЕБНОЙ ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ ПО ДИСЦИПЛИНЕ.....	10
СОДЕРЖАНИЕ ОСНОВНЫХ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ.....	13
ТЕХНОЛОГИЧЕСКАЯ КАРТА РЕЙТИНГА ДИСЦИПЛИНЫ.....	17
ФОНД ОЦЕНОЧНЫХ СРЕДСТВ.....	22
КАРТА ЛИТЕРАТУРНОГО ОБЕСПЕЧЕНИЯ ДИСЦИПЛИНЫ.....	43
КАРТА МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ ДИСЦИПЛИНЫ.....	45
ЛИСТ ВНЕСЕНИЯ ИЗМЕНЕНИЙ.....	46

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа дисциплины «Языки и методы программирования» для подготовки обучаемых по направлению 43.03.05 «Педагогическое образование» (уровень бакалавр) в рамках основной образовательной программы для профиля «Физика и информатика» разработана в соответствии с требованиями федерального государственного образовательного стандарта высшего образования 44.03.05 «Педагогическое образование» (с двумя профилями подготовки), утвержденного 09 февраля 2016 г. № 91; и рабочим учебным планом подготовки студентов КГПУ им. В.П. Астафьева по соответствующему направлению.

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

### ***Место дисциплины в структуре образовательной программы.***

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

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

***Трудоемкость дисциплины*** (общий объем времени, отведенного на изучение дисциплины):

По очной форме обучения общий объем часов – **324 (9 ЗЕТ)**, из них:

Аудиторных часов **156**:

Лекций – **78**

Практических работ – **78**

Часов самостоятельной работы – **60**

Контроль (экзамен) - **108**

### ***Цели освоения дисциплины.***

Программирование является одним из фундаментальных инструментальных методов современной информатики, поэтому учебный курс, посвященный основным методам программирования, занимает важное место в системе подготовки бакалавра и имеет как мировоззренческое, так и прикладное значение.

Ключевая цель курса - формирование системы понятий, знаний, умений и навыков в области структурного, объектно-ориентированного, функционального и логического программирования на базе языков C, C++, Lisp и Prolog. Так же, одновременно, изучаются методы проектирования, анализа и создания программного обеспечения.

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.

В данном учебном курсе рассмотрение начинается с основ программирования на языке C в среде Visual Studio, в которые включены линейные, разветвляющиеся, циклические конструкции, а также их сочетания. Также предполагается изучение простых и составных типов данных, основных принципов работы с оперативной памятью, динамических структур данных. В дальнейшем эта база используется для перехода к вопросам проектирования программного обеспечения (ПО) как с помощью новых техник программирования, так и с помощью внедрения поддерживающих инструментов в тело языка программирования, выполненного для поддержки новой парадигмы и устранения человеческого фактора при её использовании.

Курс «Языки и методы программирования» способствует углублению знаний по предметным областям:

- *Математика* (алгебра, мат. логика, и другие) – теоретическая часть курса содержит материалы, изучение которых способствует развитию математического мышления и предполагает получение некоторых дополнительных сведений по данной предметной области.
- *Моделирование* – большинство изучаемых систем и создаваемых алгоритмов неявно требуют выполнения всех ключевых этапов моделирования: абстракция, параметризация системы, выделение и формулировка связей, составление и реализация алгоритма моделирования, вычислительный эксперимент, анализ результатов и доработка модели.
- *История, Архитектура ПК* – изучение техник разработки и проектировки ПО неотделим от анализа имеющихся на данном и на

современном этапах технологиях (аппаратных и программных), их архитектуры и порождённых архитектурой возможностях и ограничениях.

- *Экономика* – рассматривается влияние информационных технологий на экономические процессы (в макро- и микроэкономике), а также детали коммерческого использования программной продукции, обеспечивающей информационную безопасность и поддержку остальных технологических процессов (включая мировые тенденции).
- *Английский язык* – используются аутентичные понятия и терминология (с приведением перевода), а также в некоторых заданиях примеры приведены на английском языке.

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Задачи освоения дисциплины	Планируемые результаты обучения по дисциплине (дескрипторы)	Код результата обучения (компетенция)
<p>Формирование культуры работы в динамичной, быстроразвивающейся области, регулируемой сообществом разработчиков и пользователей продукта в масштабах планеты.</p>	<p>Сформировать представление о языке как:</p> <ul style="list-style-type: none"> <li>- знаково-символьной системе записи выполняемых алгоритмов;</li> <li>- носителя информации о технике выполнения требуемой задачи в форме, облегчающей практику одной или нескольких существующих парадигм разработки прикладного ПО;</li> <li>- типичных конфликтах, возникающих в следствие роста сложности разрабатываемого ПО, и методиках их локализации и разрешения;</li> <li>- стандартах и процессах их разработки и внедрения.</li> </ul>	<p><b>ОК-6:</b> способность к самоорганизации и самообразованию.  <b>ПК-4:</b> способность использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых предметов.  <b>ОПК-5:</b> владение основами профессиональной этики и речевой культуры</p>
<p>Знакомство и получение первичного практического опыта проектирования и разработки ПО.</p>	<p>Выпускник владеет:</p> <ul style="list-style-type: none"> <li>- базовыми знаниями по основам структурного, ООП и функционального программирования;</li> <li>- представлением о логическом программировании и альтернативных методах программирования (машины на не архитектуре фон-Неймана и порождаемые этим последствия для языка и методики разработки);</li> <li>- основами современных математических и логико-семантических подходов в формализации, формулировании и реализации задач программирования;</li> </ul> <p>а также</p> <ul style="list-style-type: none"> <li>- имеет чёткое представление об основных методах решения программных задач;</li> <li>- способен использовать математический аппарат, методологию программирования и современные компьютерные технологии для реализации выбранных методов программирования.</li> </ul>	<p><b>ПК-4:</b> способность использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых предметов</p>

<p>Базовое обучение классических языкам программирования и методикам разработки ПО.</p>	<p>Студент должен знать:</p> <ul style="list-style-type: none"> <li>- сущность структурного, ООП и остальных подхода к программированию, его особенности и реализацию с помощью языков C, C++, 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> <li>- создание и назначение динамических структур: стеков, очередей, бинарных деревьев.</li> </ul> <p>уметь:</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>владеть навыками:</p> <ul style="list-style-type: none"> <li>- разработки алгоритмов, реализующих методы структурного программирования;</li> <li>- программирования простых алгоритмов;</li> <li>- программирования сложных алгоритмов с использованием динамических структур;</li> </ul>	<p><b>ОК-6:</b> способность к самоорганизации и самообразованию.</p> <p><b>ПК-4:</b> способность использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых предметов.</p> <p><b>ОПК-5:</b> владение основами профессиональной этики и речевой культуры</p>
---	--	---

### **Особенности технологий обучения:**

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:





Курс предполагает лекционные и лабораторные занятия.

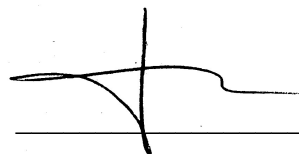
Текущий контроль осуществляется в ходе выполнения заданий на лабораторных работах, промежуточных тестах по основным модулям программы, а также итогового теста. Итоговая аттестация предусмотрена в виде зачёта/экзамена, критерием получения которого является выполнение текущего контроля и успешное прохождение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

**ЛИСТ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ С ДРУГИМИ  
ДИСЦИПЛИНАМИ НАПРАВЛЕНИЯ И ООП**

на 2016/2017 учебный год

Наименование дисциплин, изучение которых опирается на данную дисциплину	Кафедра	Предложения об изменениях в дидактических единицах, временной последовательности изучения и т.д.	Принятое решение (протокол №, дата) кафедрой, разработавшей программу
Численные методы	ИИТвО	нет	
Защита информации	ИИТвО	нет	
Компьютерное моделирование / Моделирование информационных систем	ИИТвО	нет	
Архитектура профессионального компьютера и операционные системы	ИИТвО	нет	
Теоретические основы информатики	ИИТвО	нет	
Информационные системы и сети	ИИТвО	нет	
Информационные и коммуникационные технологии в образовании	ИИТвО	нет	

Заведующий кафедрой ИИТвО



Пак Н.И.

Председатель НМС ИМФИ  
(ф.и.о., подпись)



Бортновский С.В.

05.10.2016

## ТЕХНОЛОГИЧЕСКАЯ КАРТА ОБУЧЕНИЯ ДИСЦИПЛИНЕ

ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

Направление 44.03.05 «Педагогическое образование»

Профиль: «Физика и информатика»

Квалификация: бакалавр

**Очная форма обучения**

(общая трудоемкость 9,0 з.е.)

Модули. Наименование разделов и тем	Всего часов (з.е.)	Аудиторных часов				Внеауди- торных часов	Формы и методы контроля
		всего	лекций	семина ров	лаборат. работ		
<b>Модуль 1:</b> История создания языков программирования. Парадигмы программирования	<b>0,7</b>	<b>28</b>	<b>8</b>	-	<b>8</b>	<b>12</b>	Тестирование
Тема 1.1. <b>История создания языков программирования.</b> Понятие языка программирования. Машина Бэббиджа. Низкоуровневые языки. Алгол, Фортран, С, Паскаль, С++, Java. Классификации языков программирования.	0,2	8	4	-	4	4	Собеседование
Тема 1.2. <b>Метаязыки. Грамматика языков программирования.</b> Понятия семантики, алфавита, синтаксиса языка. Язык нормальных форм Бэкуса. Синтаксические диаграммы Вирта.	0,2	8	2	-	2	4	Лабораторная работа №1
Тема 1.3. <b>Структурное программирование.</b> Сущность и свойства структурного программирования. История создания языков С и С++. Блок-схемы основных алгоритмических конструкций. Среда Visual Studio 2008	0,2	8	2	-	2	4	Лабораторная работа №2
<b>Модуль 2:</b> Базовые конструкции языка С.	<b>0,9</b>	<b>32</b>	<b>8</b>	-	<b>8</b>	<b>16</b>	
Тема 2.1. <b>Структура программы и простые типы данных в С.</b> Структура программы в С. Основные библиотеки. Простые типы	0,2	8	2	-	2	4	

данных и их характеристики. Операторы языка Переменные и константы. Инициализация. Функции ввода-вывода.							
Тема 2.2. <b>Организация ветвлений.</b> Формы записи оператора if и его применение. Простые и составные условия, логические операторы. Множественный выбор и его реализация в С.	0,2	8	2	-	2	4	
Тема 2.3. <b>Циклические конструкции</b> Синтаксис циклов с параметром, предусловием и постусловием. Их применение и взаимозаменяемость. Особенности использования цикла с параметром в С.	0,3	8	2	-	2	4	
Тема 1.4. <b>Работа с файлами в С.</b> Этапы работы с файлами в С. Основные функции работы с файлами.	0,2	8	2	-	2	4	Лабораторная работа №3
<b>Модуль 3:</b> Указатели и массивы в С	<b>1,2</b>	<b>44</b>	<b>10</b>	-	<b>12</b>	<b>22</b>	Коллоквиум
Тема 3.1. <b>Указатели.</b> Взаимодействие программы с оперативной памятью (программная модель). Понятие указателя. Объявление указателей. Адресная арифметика и особенности работы с указателями в С.	0,2	8	2	-	2	4	Лабораторная работа №4
Тема 3.2. <b>Функции.</b> Синтаксис описания функций. Понятие прототипа и применение прототипов. Формальные и фактические параметры. Способы передачи параметров в функцию: по значению, по указателю, по ссылке. Рекурсия	0,3	8	2	-	2	4	Лабораторная работа №5
Тема 3.3. <b>Статические массивы</b> Массивы в С, виды массивов. Одномерные и двумерные статические массивы. Основные этапы обработки массивов.	0,2	8	2	-	2	4	Лабораторная работа №6
Тема 3.4. <b>Динамические массивы.</b> Массивы и указатели. Операторы выделения и освобождения памяти. Создание и удаление одномерных и двумерных динамических массивов.	0,2	12	2	-	4	6	Лабораторная работа №7
Тема 3.5. <b>Строки.</b> Строки в С. Функции работы со строками. Строки, массивы и	0,3	8	2	-	2	4	Лабораторная работа №8

указатели.							
<b>Модуль 4:</b> Структуры и списки в С	<b>1,6</b>	<b>56</b>	<b>12</b>	-	<b>16</b>	<b>28</b>	Коллоквиум
Тема 4.1. <b>Структуры.</b> Понятие структуры и их назначение. Способы описания структур. Массивы структур.	0,3	12	2	-	4	6	Лабораторная работа №9
Тема 4.2. <b>Односвязные списки.</b> Списки и их назначение. Функции обработки однонаправленных списков.	0,2	8	2	-	2	4	Лабораторная работа №10
Тема 4.3. <b>Стеки и очереди.</b> Понятия стеков и очередей. Функции обработки стеков и очередей	0,2	8	2	-	2	4	Лабораторная работа №11
Тема 4.4. <b>Бинарные деревья.</b> Понятие бинарного дерева. Алгоритмы формирования и удаления бинарных деревьев	0,3	8	2	-	2	4	Лабораторная работа №12
Тема 4.5. <b>Алгоритмы обработки бинарных деревьев.</b> Обход дерева, поиск по дереву	0,3	12	2	-	4	6	Лабораторная работа №13
Тема 4.6. <b>Элементы анализа эффективности алгоритмов.</b> Подходы к анализу эффективности алгоритмов. Расчет количества операций и скорости работы программы.	0,3	8	2	-	2	4	Лабораторная работа №14
<b>Модуль 5:</b> Объектно-ориентированное программирование на С++	<b>1,1</b>	<b>40</b>	<b>10</b>	-	<b>10</b>	<b>20</b>	Коллоквиум
Тема 5.1. <b>Принципы ООП.</b> Наследование, инкапсуляция, полиморфизм. Понятия класса, экземпляра класса.	0,2	8	2	-	2	4	Лабораторная работа №15
Тема 5.2. <b>Классы.</b> Экземпляры классов. Конструкторы, деструкторы. Методы класса.	0,4	16	4	-	4	8	Лабораторная работа №16
Тема 5.3. Виртуальные и дружественные функции и их применение.	0,5	16	4	-	4	8	Лабораторная работа №17
<b>Модуль 6:</b> Основы параллельного программирования.	<b>0,9</b>	<b>34</b>	<b>8</b>	-	<b>10</b>	<b>18</b>	Коллоквиум
Тема 6.1. <b>Введение. Параллелизм. Виды параллелизма.</b>	0,2	8	2	-	2	4	Лабораторная

Принципы параллельного программирования. Перспективы и проблемы разработки параллельных алгоритмов. Примеры параллельных вычислительных систем.							работа №18
<b>Тема 6.2. Классификация параллельных вычислительных систем. Стадии разработки параллельных алгоритмов.</b> Классификация Флинна. Суперкомпьютеры и кластеры. Системы с распределенной и разделяемой памятью. Виртуальные топологии.	0,2	8	2	-	2	4	Лабораторная работа №19
<b>Тема 6.3. Основы параллельного программирования на примере библиотеки MPI для языка C++.</b> Интерфейс передачи сообщений. Классы задач, решаемые с использованием MPI. Типы передачи.	0,3	10	2	-	4	6	Лабораторная работа №20
<b>Тема 6.4. Оценка эффективности параллельных алгоритмов.</b> Максимально достижимый параллелизм. Закон Амдаля. Закон Густавсона-Барсиса. Масштабируемость алгоритма. Показатель эффективности параллельного алгоритма. Трудоемкость операция передачи данных для кластерных систем.	0,2	8	2	-	2	4	Лабораторная работа №21
<b>Модуль 7: Функциональное программирование.</b>	<b>1,2</b>	<b>44</b>	<b>10</b>	-	<b>12</b>	<b>22</b>	Коллоквиум
<b>Тема 7.1. Основы языка Лисп.</b> Принципы функционального программирования. Язык Лисп. Типы данных. Арифметические функции, функции ввода-вывода.	0,2	8	2	-	2	4	Лабораторная работа №22
<b>Тема 7.2. Лисп. Функции-распознаватели, математические функции.</b> Решение простых вычислительных задач на Лиспе. Понятие функции-распознавателя.	0,2	8	2	-	2	4	Лабораторная работа №23
<b>Тема 7.3. Лисп. Списки. Базовые функции списков.</b> Понятие списка в языке Лисп. Функции, выполняемые списками. Функции обработки списков. Список как основная структура языка.	0,2	8	2	-	2	4	Лабораторная работа №24
<b>Тема 7.4. Лисп. Условные переходы и лямбда-выражения.</b> Организация условных переходов при программировании на Лиспе. Понятие лямбда-выражения. Использование лямбда-выражений.	0,2	8	2	-	2	4	Лабораторная работа №25

Тема 7.5. <b>Лисп. Рекурсивное определение функции. Форма DEFUN.</b> Рекурсия в Лиспе и ее реализации. Форма DEFUN и ее назначение.	0,3	12	2	-	4	6	Лабораторная работа №26
<b>Модуль 8:</b> Логическое программирование.	<b>1</b>	<b>36</b>	<b>8</b>	-	<b>10</b>	<b>18</b>	Коллоквиум
Тема 8.1. <b>Пролог. Введение</b> Принципы логического программирования. Принцип работы программ на Прологе. Структура программы. Типы данных. Структуры данных. Некоторые стандартные предикаты. Поиск ответа по базе знаний.	0,2	8	2	-	2	4	Лабораторная работа №27
Тема 8.2. <b>Пролог. Поиск с возвратом. Организация повторений с помощью рекурсии</b> Принципы работы поиска с возвратом. Правила работы поиска с возвратом. Откат и отсечение. Рекурсия. Повторения с помощью рекурсии. Хвостовая рекурсия.	0,2	8	2	-	2	4	Лабораторная работа №28
Тема 8.3. <b>Пролог. Списки. Операции со списками.</b> Понятие списка. Голова и хвост списка. Рекурсивная обработка списков. Основные задачи обработки списков. Примеры.	0,3	8	2	-	2	4	Лабораторная работа №29
Тема 8.4. <b>Пролог. Решение логических задач.</b> Решение задач на графах. Решение логических задач на примере ханойских башен и расстановки ферзей.	0,3	12	2	-	4	6	Лабораторная работа №30
<b>Модуль 9:</b> Итоговый.	<b>0,3</b>	<b>10</b>	<b>2</b>	-	<b>2</b>	<b>6</b>	Тестирование
Тема 9.1. Итоговое тестирование.	0,3	10	2	-	2	6	Итоговое тестирование
<b>Итого</b>	<b>9,0 (324)</b>	<b>160</b>	<b>76</b>	-	<b>88</b>	<b>164</b>	<b>экзамен</b>

# ТЕХНОЛОГИЧЕСКАЯ КАРТА УЧЕБНОЙ ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ ПО ДИСЦИПЛИНЕ

## ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

Направление 44.03.05 «Педагогическое образование»

Профиль: «Физика и информатика»

Квалификация: бакалавр

**Очная форма обучения**

(общая трудоемкость 9,0 з.е.)

Номер модуля	Трудоемкость и сроки выполнения в часах	Содержание	Планируемые результаты	Действия проектной и исследовательской деятельности студентов	Формы и методы самоконтроля	Формы и методы контроля и оценивания
1	10	<p><u>Описательно-ознакомительные проекты:</u></p> <ol style="list-style-type: none"> <li>1. Многоядерное программирование. Системы поточной обработки. Обзор.</li> <li>2. Решение для поточной обработки OpenMP. Обзор. Необходимое программное обеспечение и его настройка</li> <li>3. Программирование видеокарт как вычислителей. Технология CUDA.</li> <li>4. Управление потоками: синхронизация и её примитивы.</li> <li>5. Проблемы поточной обработки в циклах.</li> <li>6. Распространенные проблемы параллельного программирования и пути их решения.</li> <li>7. Многопроцессорные и многоядерные системы.</li> <li>8. Распределённые системы и распределённые алгоритмы.</li> <li>9. Оценка коммуникационной трудоёмкости параллельных алгоритмов.</li> <li>10. Скалярная, векторная, конвейерная и параллельная обработка. Обзор.</li> <li>11. Параллельные формы графа алгоритма. Ярус и высота графа.</li> </ol>	<p>Сформулирована тема и определено основное содержание исследовательского реферата</p> <p>Готовность к выполнению работы</p>	<p>Для выполнения этого типа проекта вам нужно:</p> <ul style="list-style-type: none"> <li>– ознакомиться с литературой по теме;</li> <li>– выделить основные положения, определения, схемы, алгоритмы;</li> <li>– продумать методы и способы изложения сути темы сокурсникам;</li> <li>– оформить материал в виде 15-минутного доклада, сопровождаемого презентацией, метапланом, краткими формулами, ментальными картами, опорными сигналами и другими средствами, облегчающими восприятие и понимание сути доклада.</li> </ul>	<p>Самооценка на соответствие требованиям, предъявляемым к работам</p>	<p>Оценка доклада в процессе публичной защиты</p>



		<p>12. Графовые модели программ. Виды графовых моделей.</p> <p>13. Суперскалярные процессоры, процессоры с сокращённым набором команд, процессоры со сверхдлинным командным словом. Обзор</p> <p>14. Программирование с использованием PVM. Характеристики системы и особенности её работы.</p> <p>15.</p>				
2	10	<p>II. <i>Методические проекты.</i></p> <p>1. Школьникам о суперкомпьютерах.</p> <p>2. Виды и уровни параллелизма.</p> <p>3. Классификация вычислительных систем.</p> <p>4. Видеокарты как вычислители</p> <p>5. Параллельные процессы вокруг нас.</p> <p>6. Параллелизм и персональные компьютеры</p> <p>7. Управление параллельными потоками.</p> <p>8. Графовые модели программ</p> <p>9. Этапы разработки параллельной программы</p> <p>10. Языки, системы и технологии параллельного программирования</p>	<p>Разработанный урок на внеклассном мероприятии, как занятие элективного курса; в средней или старшей школе;</p>	<p>Для выполнения этого типа проекта нужно:</p> <ul style="list-style-type: none"> <li>– ознакомиться с литературой по теме;</li> <li>– выделить основные положения, определения, схемы, алгоритмы;</li> <li>– определить, в какой форме и в какой период эта тема может быть представлена школьникам: отдельным уроком, на внеклассном мероприятии, как занятие элективного курса; в средней или старшей школе;</li> <li>– разработать фрагмент урока (занятия), по объяснению нового материала, продумать методы, приемы и средства обучения школьников;</li> <li>– провести подготовленный фрагмент урока на своих сокурсниках.</li> </ul>	<p>Самооценка на соответствие требованиям, предъявляемым к работам</p>	<p>Оценка планирования по критериям обобщенного листа, разработанного во входном модуле</p>
3	10	<p>III. <i>Проекты по программированию и моделированию.</i></p> <p>1. Алгоритм Фокса умножения матриц. Модель и программа.</p> <p>2. Алгоритм Кэннона умножения матриц. Модель и программа.</p> <p>3. Параллельная сортировка Шелла (для одномерного массива).</p> <p>4. Параллельная быстрая сортировка (для одномерного массива).</p> <p>5. Задача поиска кратчайшего пути на графе. Параллельный алгоритм.</p>	<p>построенная, модель, словесный алгоритм решения задачи, со всеми этапами его</p>	<ul style="list-style-type: none"> <li>– ознакомиться с литературой по теме;</li> <li>– построить математическую модель задачи;</li> <li>– построить, опираясь на модель, словесный алгоритм решения задачи, предварительно осветив все этапы его построения и отрисовав необходимые графы;</li> <li>– предложить программную</li> </ul>	<p>Самооценка на соответствие требованиям, предъявляемым к работам</p>	<p>Оценка в процессе обсуждения на семинаре</p>

	<p>6. Численное решение задачи Дирихле для уравнения Пуассона. Метод конечных разностей. Математическая модель.</p> <p>7. Численное решение задачи Дирихле для уравнения Пуассона. Метод конечных разностей. Программная реализация модели. (Рекомендуется брать вдвоём в паре с предыдущей темой)</p> <p>8. Алгоритм Гаусса-Зейделя.</p> <p>9. Внутри кольца <math>a &lt; r &lt; b</math>, <math>0 \leq \varphi \leq 2\pi</math> найти решение <math>u(r)</math> краевой задачи <math>\Delta u(r)=0</math>, <math>u(a)=\Gamma</math>, <math>u(b)=U</math>. Построить математическую модель.</p> <p>10. Параллельный алгоритм обхода графа.</p> <p>11. Написать параллельную программу, реализующую метод Монте-Карло поиска площади фигуры, ограниченной линиями: <math>f(x) = x^2</math>; <math>f(x) = x^2/2</math> и <math>f(x) = 2x</math>.</p>	<p>построены и графы;</p>	<p>реализацию алгоритма; – описать алгоритм и процесс его построения в 10-минутном докладе.</p>		
--	---	---------------------------	---	--	--

## СОДЕРЖАНИЕ ОСНОВНЫХ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

### **Модуль 1: История создания языков программирования. Парадигмы программирования**

Тема 1.1. История создания языков программирования.

Понятие языка программирования. Машина Бэббиджа. Низкоуровневые языки. Алгол, Фортран, С, Паскаль, С++, Java. Классификации языков программирования.

Тема 1.2. Метаязыки. Грамматика языков программирования.

Понятия семантики, алфавита, синтаксиса языка. Язык нормальных форм Бэкуса. Синтаксические диаграммы Вирта.

Тема 1.3. Структурное программирование.

Сущность и свойства структурного программирования. История создания языков С и С++. Блок-схемы основных алгоритмических конструкций. Среда Visual Studio 2008

### **Модуль 2: Базовые конструкции языка С**

Тема 2.1. Структура программы и простые типы данных в С

Структура программы в С. Основные библиотеки. Простые типы данных и их характеристики. Операторы языка Переменные и константы. Инициализация. Функции ввода-вывода

Тема 2.2. Организация ветвлений

Формы записи оператора if и его применение. Простые и составные условия, логические операторы. Множественный выбор и его реализация в С.

Тема 2.3. Циклические конструкции

Синтаксис циклов с параметром, предусловием и постусловием. Их применение и взаимозаменяемость. Особенности использования цикла с параметром в С.

Тема 2.4. Работа с файлами в С

Этапы работы с файлами в С. Основные функции работы с файлами.

### **Модуль 3: Указатели и массивы в С**

Тема 3.1. Указатели

Взаимодействие программы с оперативной памятью (программная модель). Понятие указателя. Объявление указателей. Адресная арифметика и особенности работы с указателями в С

Тема 3.2. Функции

Синтаксис описания функций. Понятие прототипа и применение прототипов. Формальные и фактические параметры. Способы передачи параметров в функцию: по значению, по указателю, по ссылке. Рекурсия

Тема 3.3. Статические массивы

Массивы в С, виды массивов. Одномерные и двумерные статические массивы. Основные этапы обработки массивов.

Тема 3.4. Динамические массивы

Массивы и указатели. Операторы выделения и освобождения памяти. Создание и удаление одномерных и двумерных динамических массивов.

Тема 3.5. Строки

Строки в C. Функции работы со строками. Строки, массивы и указатели.

#### ***Модуль 4: Структуры и списки в C***

Тема 4.1. Структуры.

Понятие структуры и их назначение. Способы описания структур. Массивы структур.

Тема 4.2. Односвязные списки.

Списки и их назначение. Функции обработки однонаправленных списков.

Тема 4.3. Стеки и очереди.

Понятия стеков и очередей. Функции обработки стеков и очередей

Тема 4.4. Бинарные деревья.

Понятие бинарного дерева. Алгоритмы формирования и удаления бинарных деревьев

Тема 4.5. Алгоритмы обработки бинарных деревьев.

Обход дерева, поиск по дереву

Тема 4.6. Элементы анализа эффективности алгоритмов.

Подходы к анализу эффективности алгоритмов. Расчет количества операций и скорости работы программы.

#### ***Модуль 5: Объектно-ориентированное программирование на C++***

Тема 5.1. Принципы ООП.

Наследование, инкапсуляция, полиморфизм. Понятия класса, экземпляра класса.

Тема 5.2. Классы. Экземпляры классов. Конструкторы, деструкторы. Методы класса.

Тема 5.3. Виртуальные и дружественные функции и их применение.

#### ***Модуль 6: Основы параллельного программирования***

Тема 6.1. Введение. Параллелизм. Виды параллелизма.

Принципы параллельного программирования. Перспективы и проблемы разработки параллельных алгоритмов. Примеры параллельных вычислительных систем.

Тема 6.2. Классификация параллельных вычислительных систем. Стадии разработки параллельных алгоритмов.

Классификация Флинна. Суперкомпьютеры и кластеры. Системы с распределенной и разделяемой памятью. Виртуальные топологии.

Тема 6.3. Основы параллельного программирования на примере библиотеки MPI для языка C++.

Интерфейс передачи сообщений. Классы задач, решаемые с использованием MPI. Типы передачи.

Тема 6.4. Оценка эффективности параллельных алгоритмов.

Максимально достижимый параллелизм. Закон Амдаля. Закон Густавсона-Барсиса. Масштабируемость алгоритма. Показатель эффективности параллельного алгоритма. Трудоемкость операция передачи данных для кластерных систем.

### ***Модуль 7: Функциональное программирование***

Тема 7.1. Основы языка Лисп.

Принципы функционального программирования. Язык Лисп. Типы данных. Арифметические функции, функции ввода-вывода.

Тема 7.2. Лисп. Функции-распознаватели, математические функции.

Решение простых вычислительных задач на Лиспе. Понятие функции-распознавателя.

Тема 7.3. Лисп. Списки. Базовые функции списков.

Понятие списка в языке Лисп. Функции, выполняемые списками. Функции обработки списков. Список как основная структура языка

Тема 7.4. Лисп. Условные переходы и лямбда-выражения.

Организация условных переходов при программировании на Лиспе. Понятие лямбда-выражения. Тема 7.5. Лисп. Рекурсивное определение функции. Форма DEFUN.

Рекурсия в Лиспе и ее реализации. Форма DEFUN и ее назначение. Использование лямбда-выражений.

### ***Модуль 8: Логическое программирование***

Тема 8.1. Пролог. Введение

Принципы логического программирования. Принцип работы программ на Прологе. Структура программы. Типы данных. Структуры данных. Некоторые стандартные предикаты. Поиск ответа по базе знаний.

Тема 8.2. Пролог. Поиск с возвратом. Организация повторений с помощью рекурсии

Принципы работы поиска с возвратом. Правила работы поиска с возвратом. Откат и отсечение. Рекурсия. Повторения с помощью рекурсии. Хвостовая рекурсия.

Тема 8.3. Пролог. Списки. Операции со списками.

Понятие списка. Голова и хвост списка. Рекурсивная обработка списков. Основные задачи обработки списков. Примеры.

Тема 8.4. Пролог. Решение логических задач.

Решение задач на графах. Решение логических задач.

### ***Модуль 9: Итоговый***

Тема 9.1. Итоговое тестирование (зачёт и/или экзамен).

## ТЕХНОЛОГИЧЕСКАЯ КАРТА РЕЙТИНГА ДИСЦИПЛИНЫ

<b>Наименование дисциплины</b>	<b>Направление подготовки и уровень образования (бакалавриат, магистратура, аспирантура) Наименование программы/ профиля</b>	<b>Количество з.е.</b>
Языки и методы программирования	НАПРАВЛЕНИЕ: 44.03.05 «Педагогическое образование» Профиль: «Физика и информатика» Квалификация (степень): бакалавр по очной форме обучения	<b>9</b>
<b>Смежные дисциплины по учебному плану</b>		
<b>Предшествующие:</b>		
«Информатика»		
<b>Последующие:</b>		
«Численные методы», «Защита информации», «Компьютерное моделирование / Моделирование информационных систем», «Архитектура профессионального компьютера и операционные системы», «Теоретические основы информатики», «Информационные системы и сети», «Информационные и коммуникационные технологии в образовании»		

<b>БАЗОВЫЙ МОДУЛЬ № 1</b>			
	Форма работы	Количество баллов 5 %	
		min	max
Текущая работа	Доклад	1	1
	Домашнее задание	1	1
	Лабораторная работа (аудиторная)	1	1
Промежуточный рейтинг-контроль	Тестирование №1	2	2
Итого		<b>5</b>	<b>5</b>

<b>БАЗОВЫЙ МОДУЛЬ № 3</b>			
	Форма работы	Количество баллов 15 %	
		min	max
Текущая работа	Доклад	2	3
	Домашнее задание	2	2
	Лабораторная работа (аудиторная)	2	3
Промежуточный рейтинг-контроль	Коллоквиум №1	4	7
<b>Итого</b>		<b>10</b>	<b>15</b>

<b>БАЗОВЫЙ МОДУЛЬ № 4</b>			
	Форма работы	Количество баллов 15 %	
		min	max
Текущая работа	Доклад	1	3
	Домашнее задание	1	2
	Лабораторная работа (аудиторная)	1	3
Промежуточный рейтинг-контроль	Коллоквиум №2	3	7
<b>Итого</b>		<b>6</b>	<b>15</b>

<b>БАЗОВЫЙ МОДУЛЬ № 5</b>			
	Форма работы	Количество баллов 10 %	
		min	max
Текущая работа	Доклад	1	2
	Домашнее задание	1	1
	Лабораторная работа (аудиторная)	1	2
Промежуточный рейтинг-контроль	Тестирование №1	2	5
<b>Итого</b>		<b>5</b>	<b>10</b>

<b>БАЗОВЫЙ МОДУЛЬ № 6</b>			
	Форма работы	Количество баллов 15 %	
		min	max
Текущая работа	Доклад	2	3
	Домашнее задание	2	2
	Лабораторная работа (аудиторная)	2	3

Промежуточный рейтинг-контроль	Коллоквиум №3	4	7
Итого		<b>10</b>	<b>15</b>

<b>БАЗОВЫЙ МОДУЛЬ № 7</b>			
	Форма работы	Количество баллов 10 %	
		min	max
Текущая работа	Доклад	2	2
	Домашнее задание	2	1
	Лабораторная работа (аудиторная)	2	2
Промежуточный рейтинг-контроль	Коллоквиум №4	4	5
Итого		<b>10</b>	<b>10</b>

<b>БАЗОВЫЙ МОДУЛЬ № 8</b>			
	Форма работы	Количество баллов 15 %	
		min	max
Текущая работа	Доклад	3	3
	Домашнее задание	2	2
	Лабораторная работа (аудиторная)	2	3
Промежуточный рейтинг-контроль	Тестирование №3	3	7
Итого		<b>10</b>	<b>15</b>

<b>ИТОГОВЫЙ МОДУЛЬ</b>		
Форма работы	Количество баллов	
	min	max
Тестирование	5	15
Итого	5	15

<b>ДОПОЛНИТЕЛЬНЫЙ МОДУЛЬ</b>		
Форма работы	Количество баллов	
	min	max
Работа над проектом	0	15
Итого	0	15
<b>Общее количество баллов по дисциплине (по итогам изучения)</b>	<b>60</b>	<b>100</b>



<b>всех модулей, без учета дополнительного модуля)</b>		
--	--	--

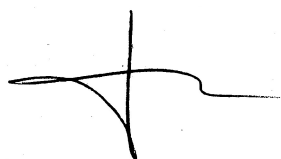
**Соответствие рейтинговых баллов и академической оценки**

<i>Общее количество набранных баллов</i>	<i>Академическая оценка</i>
меньше <b>60</b> или незакрытый модуль	<b>не зачтено</b>
	<b>зачтено</b>

ФИО преподавателя: Романов Дмитрий Валерьевич

Утверждено на заседании кафедры «05» октября 2016 г. Протокол № 3

Зав. кафедрой



Н.И. Пак

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

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 № 3

от «05» октября 2016 г.



ОДОБРЕНО

на заседании научно-методического  
совета направления подготовки

Протокол № 2

от «26» октября 2016 г.



## **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

для проведения текущего контроля и промежуточной аттестации  
обучающихся

### **«Языки и методы программирования»**

(наименование дисциплины/модуля/вида практики)

### **44.03.05 «Педагогическое образование»**

(код и наименование направления подготовки)

### **Профиль «Физика и информатика»**

(наименование профиля подготовки/наименование магистерской программы)

### **бакалавр**

(квалификация (степень) выпускника)

Составитель: Романов Д.В., доцент кафедры ИИТО

## **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 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» и его филиалах.

**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-5** - владение основами профессиональной этики и речевой культуры.

*в) профессиональные:*

**ПК-4** - способностью использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов

обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых предметов.

## 2.2. Этапы формирования и оценивания компетенций

Компетенции формируются на раннем этапе с помощью учебного научно-исследовательского проекта (УНИП), темы которого даны в соответствующем разделе, оттачиваются при выполнении соответствующей практической работы (больше 30-ти лабораторных работ), апробируются и тестируются при написании реферата и сдаче зачёта/экзамена. Ниже приведены компетенции и соответствующие КИМ.

Компетенция	Этап формирования компетенции	Дисциплины, практики, участвующие в формировании компетенции	Тип контроля	Оценочное средство / КИМы	
				Номер	Форма
<b>ОПК-5</b> - владение основами профессиональной этики и речевой культуры	ориентировочный	Языки и методы программирования	текущий контроль	1	УНИП
	когнитивный	Языки и методы программирования	текущий контроль	2	практическая работа
	праксиологический	Языки и методы программирования	Промежуточная аттестация	3	реферат
	рефлексивно-оценочный	Языки и методы программирования	Промежуточная аттестация	4	зачёт/экзамен
<b>ОК-6</b> - способность к самоорганизации и самообразованию	ориентировочный	Языки и методы программирования	текущий контроль	1	УНИП
	когнитивный	Языки и методы программирования	текущий контроль	2	практическая работа
	праксиологический	Языки и методы программирования	Промежуточная аттестация	3	реферат
	рефлексивно-оценочный	Языки и методы программирования	Итоговая аттестация	4	зачёт/экзамен

<b>ПК-4</b> - способностью использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых предметов ;	ориентировочный	Языки и методы программирования	текущий контроль	1	УНИП
	когнитивный	Языки и методы программирования	текущий контроль	2	практическая работа
	праксиологический	Языки и методы программирования	Промежуточная аттестация	3	реферат
	рефлексивно-оценочный	Языки и методы программирования	Итоговая аттестация	4	зачёт/экзамен

### 3. Фонд оценочных средств для промежуточной аттестации

Фонды оценочных средств включают:

3.1. Перечень вопросов к зачёту/экзамену/коллоквиуму.

3.2. Отдельные лабораторные работы (вынесены в электронном виде в отдельный модуль Moodle на сайте университета из-за большого объёма).

3.3. Темы УНИП (вынесены в отдельный раздел, с технологической картой).

Критерии оценивания:

Формируемые компетенции	Высокий уровень сформированности компетенций	Продвинутый уровень сформированности компетенций	Базовый уровень сформированности компетенций
	(20 - 23 балла) отлично	(16 - 19 баллов) хорошо	(13 - 15 баллов)* Удовлетворительно
<b>ОК-6</b> - способность к самоорганизации и самообразованию	Обучающийся грамотно планирует бюджет времени и других ресурсов, свободно использует инструменты и методики самоорганизации (GTD, Pomodoro, SWAT анализ). Обучающийся способен выделять собственные дефициты, искать качественные источники знаний, обучаться самостоятельно.	Обучающийся способен оценивать бюджет времени и ресурсов, имеет понятие о инструментах и методиках самоорганизации. Обучающийся способен выделять и формулировать собственные дефициты, искать источники знаний для их заполнения.	Обучающийся имеет понятие о методиках самоорганизации и управления временем, способен выделять и конструктивно формулировать собственные дефициты. Имеет представление о методике самообучения.
<b>ОПК-5</b> - владение	Обучающийся	Обучающийся	Обучающийся

основами профессиональной этики и речевой культуры	демонстрирует высокий уровень владения, способен вести дискуссию, изолировать и анализировать противоречия в споре, формулировать ключевые вопросы для разрешения противоречий.	демонстрирует хороший уровень владения, способен вести дискуссию, формулировать свою точку зрения в доступной и ясной форме.	демонстрирует способности к речевой культуре, способен ясно понимать профессиональную речь и изъясняться с использованием соответствующего понятийного и речевого аппарата с соблюдением принятых культурных и профессиональных норм.
<b>ПК-4</b> - способностью использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых предметов;	Имеет полное представление о теоретическом и практическом аппарате дисциплины, имеет базу примеров применения её на производстве и в НИР, способен пополнять её самостоятельно с предварительным выделением дефицитов. Способен проектировать любое количество примеров и задач любого уровня сложности.	Имеет представление о теоретическом и практическом аппарате дисциплины, получил базу примеров применения её на производстве и в НИР. Способен проектировать примеры и задачи с акцентом на требуемый материал. Имеет представление о методическом аппарате и состоянии литературы в области.	Ознакомлен с теоретическим и практическим аппарате дисциплины. Получил базу примеров применения её на производстве и в НИР. Способен проектировать примеры и задачи с акцентом на требуемый материал. Ознакомлен с методическим аппаратом и состоянием литературы в области.

\*Менее 13 баллов – компетенция не сформирована

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

Фонды оценочных средств включает:

1. Лабораторные работы (вынесены в электронном виде в отдельный модуль Moodle на сайте университета из-за большого объёма).  
Стоимость в баллах каждой лабораторной работы и посещения занятия приведены в соответствующей электронной таблице.
2. Контрольное тестирование, проводимое по сети с помощью учебного ресурса, уже размещённого на сайте университета.

#### **5. Учебно-методическое и информационное обеспечение фондов оценочных средств (см. карту литературного обеспечения дисциплины).**

### **КИМ: входной контроль уровня знаний.**

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



15. Отметьте правильное определение константы:
1. `const long BITS = 32;`
  2. `const bit ZERO = 0x0;`
  3. `const float 0 Ora = 5.9787;`
16. Укажите пункты, в которых происходит объявление констант и переменных
1. `float dD;`
  2. `float sd2 = 3.2;`
  3. `bType = 3;`
  4. `int k; k = 89;`
17. Укажите, в каких выражениях используются ключевые слова?
1. `sdf = 2; int r = 24;`
  2. `TStringList *S = new TStringList;`
  3. `x = 3; x = x + 4;`
  4. `void function()`
18. Если после выражения стоит точка с запятой, то
1. это оператор-выражение, действие которого заключается в вычислении выражения
  2. выражение вычисляется, а его значение запоминается в специальной переменной, которую можно использовать в следующем операторе
  3. выражение вычисляется только если первой стоит операция присваивания
19. Чему равен результат вычисления выражения  $x + 3 * b + x$  при  $x = 12$  и  $b = 8$  ?
1. 132
  2. 48
  3. 300
20. Каково будет значение переменной  $k$  после выполнения оператора `k = ++ k`; если до его выполнения  $k$  равнялось 6?
1. 6
  2. 7
  3. 8
21. 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";
```
1. aaa
  2. bbb
  3. ccc
  4. aaacc
  5. bbbccc
  6. ошибка компиляции
22. Что выведет на экран следующая программа ?
- ```
# include < iostream. h>
int main () {
int i;
for(i = 0; i < 9; i++)
cout << i+1;
return 1;
}
```
1. цифры от 0 до 8
  2. цифры от 1 до 9
  3. программа не будет построена из-за ошибок
23. Укажите, какой будет результат вычисления  $k$ ?
- ```
int func()
{
```

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

1. 0
2. 10
3. 1
4. бесконечный цикл
5. 100

24. Если int n=45, какой будет результат?

```
switch(n) {
case 23: cout << "aaa";
case 45: cout << "bbb";
default: cout << "vvv";
break; }
```

1. ошибка компилятора
2. aaa
3. ббб
4. vvv
5. бббvvv
6. aaavvv
7. неопределенное поведение

25. 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:
```

1. Вариант 1
2. Вариант 2
3. Вариант 3
4. ничего не будет напечатано
5. программа не откомпилируется

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

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

1. 0
2. 10
3. 1
4. бесконечный цикл
5. 100

27. Если `int n=3`, какой будет результат?

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

1. ошибка компилятора
2. aaa
3. ббб
4. ввв
5. неопределенное поведение

28. Что произойдет при выполнении:

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

1. функция `func` выполнится 5 раз
2. функция `func` не выполнится ни разу
3. функция `func` будет выполняться бесконечно

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

1. `if (...) then ... else ...`
2. `while (...) {...}`
3. `switch (...) {...}`
4. `for (...; ...; ...) {...}`

30. Укажите результат вычисления цикла:

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

1. `k = 2;`
2. `k = 0;`
3. `k = 5;`
4. `k = 4;`

31. Какой из наборов перечисляемых значений записан правильно?

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

32. Каким будет результат работы следующего фрагмента программы:

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

1. 63.50
2. 64.00
3. 3.50
4. 4.00
5. 63.00

33. Найдите недопустимую запись символьной константы:

1. `'F'`
2. `"\022"`
3. `'Ю'`
4. `'\ a'`

34. Какой результат вычисления следующего выражения?

1.  $0 \text{xFF} \& 5 \gg 1 + 1$
2. 1
3. 2
4. 256

35. Каким будет результат следующей программы:

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

1. 8.5
2. 9.0
3. 8.0
4. 9.5
5. 7.5

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

1. они существуют только во время выполнения функции
2. к ним можно обращаться только после их объявления
3. их имена должны начинаться с символа l

37. Отметьте допустимые имена функций:

1. `_This_Function_12_x_`
2. `ax%u7`
3. `ffffffAAAAAA`
4. `3_pi`
5. `calculateIt`
6. `thisname_is_too_long_forafunction`

38. Как называется функция, которая вызывает саму себя?

1. конструктор
2. деструктор
3. подставляемая
4. рекурсивная

39. В чем разница между фактическими и формальными параметрами?

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

40. Укажите правильное объявление функции

1. `int MyFunc(double x, y, int i)`
2. `void correct(double d = 3.14, double a)`
3. `int sum(int j, int k, int x = 0)`

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

1. оператор return должен стоять последним в теле функции
2. оператор return завершает выполнение функции
3. в теле функции должен присутствовать только один оператор return

42.39. 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:

1. A := X^.Next^.Next
2. X := Y
3. Logic := X^.X
4. X^.Next := A
5. X^ := Y^.Next

43.40. 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;

```

End;

выполняется

1. перемещение компонента к началу списка;
2. сортировка компонентов списка в порядке возрастания;
3. сортировка компонентов списка в порядке убывания;
4. перестановка соседних компонентов списка;
5. добавление в список нескольких новых компонент.

44. 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;

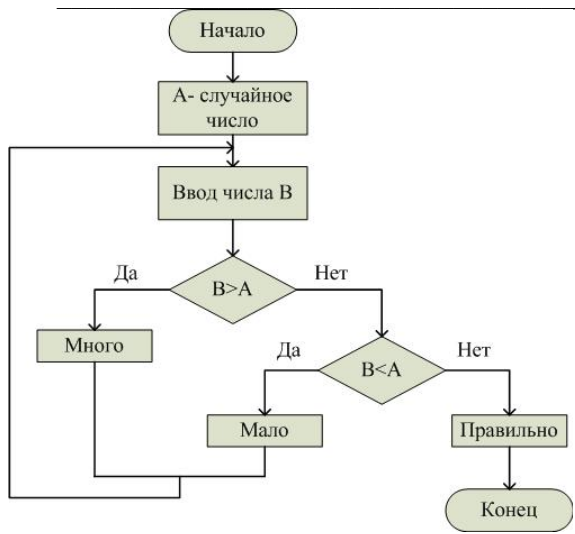
```

определяется

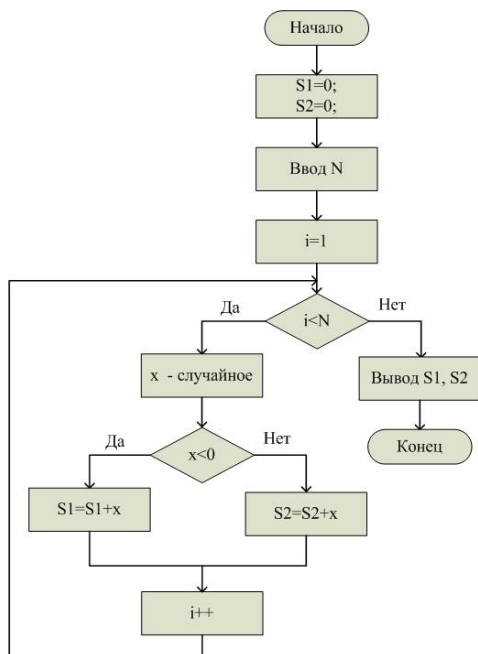
1. первый элемент списка;
  2. сумма элементов списка;
  3. сумма первого и последнего элементов списка;
  4. количество элементов списка;
  5. количество звеньев списка, где указатель на следующее звено не Nil.
45. Даны натуральные числа  $a$ ,  $b$ , обозначающие соответственно числитель и знаменатель дроби. Сократить дробь, т.е. получить правильную несократимую дробь. Какую часть в предложенной задаче вы можете выделить как функцию?
- 

46. 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?
- 

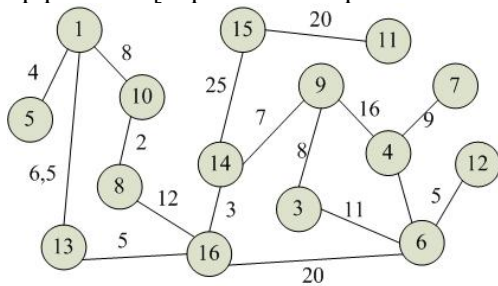
47. Проанализировав схему, предположите, решение какой задачи отображено.
- 



48. Проанализировав схему, предположите, решение какой задачи отображено.
-



49. 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.



**КИМ: итоговый контроль знаний и уровня сформированности параллельного стиля мышления.**

**(при составлении теста использовались материалы портала <http://sigma.parallel.ru>)**

1. Изображённая вычислительная сеть – это:
  1. решётка;
  2. звезда;
  3. тор;
  4. линейка;
2. На рисунке – сеть типа:
  1. тор;
  2. полная связь;
  3. решётка;
  4. кольцо;
3. На рисунке – сеть типа:
  1. полная связь;
  2. кольцо;
  3. решётка;
  4. линейка;
4. Отметьте, какие утверждения о массивно-параллельных компьютерах верны?
  1. массивно-параллельные компьютеры не могут работать без хост-машины;
  2. основными отличительными характеристиками массивно-параллельных компьютеров являются: используемые микропроцессоры и коммуникационная сеть (среда);
  3. в массивно-параллельных компьютерах не может быть больше 1024 процессоров;
  4. коммуникационная сеть массивно-параллельных компьютеров всегда организуется в виде трёхмерного тора.
5. Отметьте верные утверждения о кластерных вычислительных системах:
  1. кластерные системы строятся на базе серийных процессоров;
  2. на каждом узле кластера выполняется свой экземпляр операционной системы;
  3. максимально возможное число процессоров кластерной системы равно 128;
  4. один кластер может строиться с использованием нескольких коммуникационных технологий.
6. Разновидностью распараллеливания являются технологии и приёмы:
  1. суперскалярности;
  2. структурного программирования;
  3. многопроцессорности;
  4. объектно-ориентированного программирования.
7. Отметьте верные утверждения:
  1. MPI - это сокращение от Message Passing Interface.
  2. Функция MPI\_Comm\_size определяет общее число запущенных параллельных процессов приложения.
  3. Каждый параллельный процесс в MPI имеет номер.
  4. MPI - это сокращение от Multiple Parallel Interface.
8. Какая операция MPI не относится к коллективным операциям?
  1. MPI\_Bcast;
  2. MPI\_Send;
  3. MPI\_Reduce;
  4. MPI\_Gather.
9. Отметьте верные утверждения:
  1. использование функций MPI\_Send и MPI\_Recv может привести к тупиковой ситуации (deadlock);
  2. в коллективных операциях участвуют все процессы приложения;
  3. функция, соответствующая коллективной операции, должна быть вызвана каждым процессом, быть может, со своим набором параметров;



4. возврат процесса из функции, реализующей коллективную операцию, всегда означает, что операция уже завершена;
10. Фрагмент программы:

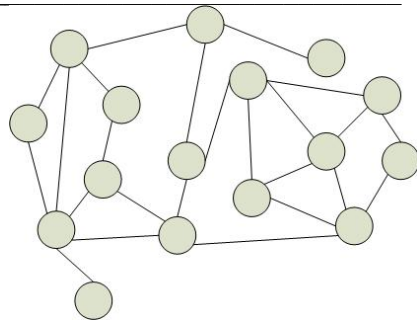
```
MPI_Comm_size(comm, &size);
MPI_Send( buf, 15, MPI_INT, size+1, 8, comm);
MPI_Send( buf, 8, MPI_INT, size-1, 15, comm);
```

  1. может быть правильным;
  2. заведомо содержит одну ошибку;
  3. заведомо содержит две ошибки;
  4. заведомо содержит три ошибки.
11. Отметьте верные утверждения о функции MPI\_Recv:
  1. перед вызовом функции MPI\_Recv надо обратиться к функции MPI\_Get\_count;
  2. возврат из функции означает, что либо произошла ошибка, либо принятое сообщение расположено в первом параметре;
  3. функцией MPI\_Recv нельзя принимать сообщение, посланное с помощью функции MPI\_Ssend;
  4. нельзя использовать функцию MPI\_Recv, если неизвестны отправитель сообщения или тег сообщения.
12. Посылка сообщения с блокировкой (MPI\_Send) означает, что возврат из функции произойдет тогда, когда:
  1. можно повторно использовать параметры данной функции;
  2. сообщение покинет процесс;
  3. сообщение принято адресатом;
  4. адресат инициировал приём данного сообщения.
13. Функция MPI\_Send возвращает:
  1. число байт, заявленное в вызове для пересылки;
  2. значение MPI\_SUCCESS или код ошибки;
  3. номер процесса, которому адресована передача;
  4. число реально переданных байт.
14. В декартовой топологии множество процессов представляется в виде:
  1. прямоугольной решётки;
  2. графа произвольного вида;
  3. полного графа;
  4. звезды.
15. MPI поддерживает топологии вида:
  1. прямоугольная решётка произвольной размерности и граф произвольного вида;
  2. только прямоугольная решётка произвольной размерности;
  3. только граф произвольного вида;
  4. тор произвольной размерности и граф произвольного вида.
16. Операция широковещательной рассылки данных это:
  1. операция рассылки значений ведущим процессом всем остальным процессам коммутатора; все процессы получают часть исходных данных;
  2. операция рассылки значений ведущим процессом всем остальным процессам коммутатора, все процессы получают рассылаемые данные целиком;
  3. операция рассылки различающихся значений ведущим процессом всем остальным процессам коммутатора;
  4. операция рассылки значений ведущим процессом всем остальным процессам приложения, все процессы получают рассылаемые данные целиком.
17. Операцию редукции данных MPI\_Reduce можно описать:
  1. операцию передачи данных, при которой над собираемыми значениями осуществляется та или иная обработка, при этом результат обработки получают все процессы коммутатора;
  2. как операцию передачи данных, при которой над собираемыми значениями осуществляется обработка, при этом частичные значения результатов редуцирования получают все процессы параллельной программы;

3. как операцию передачи данных, при которой над собираемыми значениями осуществляется обработка в процессе передачи, при этом результат обработки получает только ведущий процесс;
  4. операция передачи данных, при которой все процесса коммуникатора получают различающиеся значения.
18. В коллективных операциях передачи данных обязаны принимать участие:
    1. все процессы программы;
    2. все процессы группы коммуникаторов;
    3. все процессы одного коммуникатора;
    4. некоторые процессы одного коммуникатора.
  19. Режим передачи по готовности может быть использован только если:
    1. операция приёма сообщения уже инициирована;
    2. операция приёма сообщения гарантированно будет запущена позднее момента начала передачи сообщения;
    3. при размере сообщения, меньшем размера системного буфера.
  20. В буферизованном режиме функция отправки сообщения завершается:
    1. сразу же после копирования сообщения в системный буфер;
    2. при получении от процесса-получателя подтверждения о начале приёма отправленного сообщения;
    3. при начале фактической передачи сообщения.
  21. В синхронном режиме передачи завершение функции отправки сообщения происходит:
    1. при старте передачи данных процессом-отправителем по сети;
    2. при завершении копирования сообщения в системный буфер;
    3. 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.
  22. Приём сообщений при помощи функции `MPI_Recv` может быть выполнен:
    1. от любого адресата и с любым тегом при указании специальных значений в качестве параметров вызова функции,
    2. от любого адресата, однако, тег сообщения должен быть указан однозначно,
    3. только от однозначно определяемого адресата с заданным тегом.
  23. Прием сообщения при помощи функции `MPI_Recv` может быть инициирован:
    1. только до момента начала отправки сообщения;
    2. только после момента начала отправки сообщения;
    3. до момента, в момент или после момента начала отправки сообщения;
    4. только в момент начала отправки сообщения.
  24. Все данные для передачи в качестве сообщения `MPI` описываются с помощью триады:
    1. адрес памяти, ранг процесса-отправителя, используемый коммуникатор,
    2. адрес памяти, ранг процесса-получателя, используемый коммуникатор.
    3. адрес памяти, количество и тип элементов данных,
  25. Функция `MPI_Recv`:
    1. в зависимости от используемой операции передачи может как заблокировать, так и не заблокировать процесс-получатель.
    2. блокирует процесс-получатель до момента фактического получения сообщения,
    3. принимает сообщение в фоновом режиме, процесс в это время может продолжать вычисления.
  26. Указание используемого коммуникатора является:
    1. обязательным для всех операций передачи данных в `MPI`,
    2. необязательным для некоторых операций передачи данных в `MPI`,
    3. обязательным для некоторых операций передачи данных в `MPI`.
  27. Под коммуникатором в `MPI` понимается:
    1. группа процессов, в рамках которой выполняются операции передачи данных,
    2. пара процессов, в рамках которой происходит информационное взаимодействие.
    3. специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров, используемых при выполнении операций передачи данных,
  28. Под параллельной программой в рамках `MPI` понимается:

1. множество одновременно работающих процессоров.
  2. множество одновременно выполняемых процессов,
  3. множество одновременно выполняемых потоков,
29. Процессы параллельной программы в рамках MPI:
1. могут выполняться на разных процессорах, на одном процессоре могут располагаться несколько процессов,
  2. обязательно выполняются на одном процессоре.
  3. могут выполняться только на разных процессорах,
30. Завершение вызова функции неблокирующего обмена означает:
1. фактическое выполнение обмена данными;
  2. фактическое выполнение приема данных (для функции неблокирующего приема) или начало фоновой передачи (для функции неблокирующей передачи);
  3. инициацию запрошенной операции передачи, но ничего не говорит о начале или завершённости обмена.
31. Дана матрица размером  $N \times M$ . Написать программу транспонирования матрицы. Предложите способ разбиения задачи на подзадачи, опишите информационные связи между подзадачами.
- 
- 

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



- 1.
33. 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.
- 
- 

34. Как передать с 1-го процесса на 8-й целое число 5 в рамках коммуникатора MPI\_COMM\_WORLD?
- 

35. Как передать с 3-го процесса на 0-й массив вещественных чисел из 10 элементов в рамках коммуникатора MPI\_COMM\_WORLD?
- 

36. Как передать с 0-го процесса всем остальным массив из 20 символов в рамках коммуникатора MPI\_COMM\_WORLD?
- 

37. 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?
- 

38. 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.
- 
-

39. На 15 процессах коммуникатора MPI\_COMM\_WORLD вычислены промежуточные значения. Напишите функцию, которая найдёт максимальное значение и передаст результат на 0-й процесс. \_\_\_\_\_
40. 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. \_\_\_\_\_

### **КИМ: вопросы к коллоквиуму**

1. История создания языков программирования. 19 век.
2. История создания языков программирования. Первые языки. 40-е гг 20 века.
3. История создания языков программирования. Ассемблер и первые языки высокого уровня.
4. История создания языков программирования. Языки процедурного программирования (PL/1, Pascal, Ada).
5. История создания языков программирования. Языки процедурного программирования (C)
6. История создания языков программирования. Языки функционального программирования (Лисп)
7. История создания языков программирования. Языки логического программирования (Пролог)
8. История создания языков программирования. Языки объектно-ориентированного программирования (C++, Java)
9. Классификации языков программирования.
10. Метаязыки. Алфавит, синтаксис, семантика и понятие языков программирования.
11. Язык БНФ. Основные концепции. Пример.
12. Синтаксические диаграммы Вирта. Правила составления. Пример.
13. Грамматика языков программирования. Основные понятия.
14. Основные функции микропроцессора.
15. Регистры общего назначения.
16. Регистры сегментов, указатель инструкций и регистр флагов.
17. Управляющие регистры, регистры системных адресов.
18. Работа микропроцессора с памятью.
19. Основные команды языка ассемблер. Пример. Основные рабочие регистры, используемые в программах на ассемблере.
20. Пакет MASM32. Состав пакета. Установка пакета. Основные файлы. Компиляция программ.
21. Структура программы в MASM32 (пояснить каждый элемент структуры).
22. Директивы include, includelib, DUP, invoke. Примеры использования
23. Типы данных. Примеры определения данных
24. Символические имена. Примеры.

25. Логические структуры языка программирования. Организация условных переходов и ветвлений. Команды условных и безусловных переходов. Пример.
26. Логические команды ассемблера.
27. Организация циклов в ассемблере. Команда LOOP и ее модификации. Примеры.
28. Операции обработки строк. Идентификация строк.
29. Команды строковых примитивов, их действие. Пример.
30. Команды CMOVE, XCHG, XLAT, LEA, PTR, SHR
31. Организация подпрограмм, размещение подпрограмм. Синтаксис.
32. Работа стека. Пример.
33. Параметры процедур и возвращаемые значения.

### **Языки и методы программирования.**

34. История создания языков программирования. 19 век. Первые языки. 40-е гг 20 века. Ассемблер и первые языки высокого уровня.
35. История создания языков программирования. Языки процедурного программирования (PL/1, Pascal, Ada, C), языки функционального программирования (Лисп), языки логического программирования (Пролог)
36. История создания языков программирования. Языки объектно-ориентированного программирования (C++, Java). Классификации языков программирования
37. Метаязыки. Алфавит, синтаксис, семантика и понятие языков программирования. Язык БНФ. Основные концепции.
38. Синтаксические диаграммы Вирта. Правила составления. Грамматика языков программирования. Основные понятия. Примеры.
39. Основные функции микропроцессора.
40. Регистры общего назначения. Регистры сегментов, указатель инструкций и регистр флагов.
41. Управляющие регистры, регистры системных адресов. Работа микропроцессора с памятью.
42. Основные команды языка ассемблер. Пример. Основные рабочие регистры, используемые в программах на ассемблере.
43. Пакет MASM32. Состав пакета. Установка пакета. Основные файлы. Компиляция программ. Структура программы в MASM32 (пояснить каждый элемент структуры).
44. Директивы include, includelib, DUP, invoke. Примеры использования. Типы данных. Примеры определения данных
45. Символические имена. Примеры. Логические структуры языка программирования. Организация условных переходов и ветвлений. Команды условных и безусловных переходов. Пример.

46. Логические команды ассемблера. Организация циклов в ассемблере. Команда LOOP и ее модификации. Примеры.
47. Операции обработки строк. Идентификация строк. Команды строковых примитивов, их действие. Пример.
48. Команды CMOVE, XCHG, XLAT, LEA, PTR, SHR.
49. Организация подпрограмм, размещение подпрограмм. Синтаксис.
50. Работа стека. Пример. Параметры процедур и возвращаемые значения.
51. Язык C/C++. Динамические переменные. Указатели. Описание указателя, операторы для работы с указателями. Адресная арифметика. Массивы и указатели. Примеры.
52. Язык C/C++. Динамические списки. Стеки и очереди. Примеры процедур обработки списков.
53. Пролог. Принципы логического программирования. Высказывания и предикаты. Работа программ на Прологе. Области применения языка.
54. Структура программы на Прологе. Назначение каждого раздела. Синтаксис. Типы данных в Прологе. Примеры.
55. Предикаты, примеры описания предикатов. Пустые предикаты. Стандартные предикаты.
56. Поиск с возвратом. Его особенности. Пример. Пояснить на примере ход поиска с возвратом. Поиск всех альтернативных решений. Правила поиска с возвратом.
57. Отсечение и откат. Примеры.
58. Рекурсия. Случаи применения рекурсии. Откат с петлями (конструкция `gereat. gereat :- gereat`). Пример. Применение рекурсии для вычислений. Пример. Схема работы рекурсивной процедуры. Понятие стекового фрейма. Хвостовая рекурсия.
59. Списки. Два определения списка. Примеры определения списков. Обработка списков, этапы обработки. Основные задачи обработки списков. Примеры реализации перечисленных задач (по крайней мере трех).
60. Решение логических задач на Прологе.
61. Функциональное программирование. Принципы функционального программирования. Язык Лисп и его диалекты. Атомы и S-выражения. Списки и способы их записи. Функции присваивания и ввода-вывода данных. Примеры.
62. Язык Лисп. Математические функции. Числовые предикаты. Функции работы с данными. Примеры.
63. Язык Лисп. Основные конструкции: переменные, константы, функции, ветвления (IF, COND). Примеры
64. Язык Лисп. Рекурсивные функции. Пример. Механизм обработки программ, представленных S-выражениями.

65. Язык Лисп. Основные управляющие конструкции: оператор IF, конструкции WHEN и UNLESS. Конструкции AND, OR, NOT. Примеры.
66. Язык Лисп. Организация циклов: циклы DOLIST и DOTIMES, конструкция DO. Примеры.
67. Язык Лисп. Организация циклов. Цикл LOOP, ключевые слова цикла LOOP. Возврат значения из цикла LOOP. Примеры.
68. Язык Лисп. Функции. Именованные и безымянные функции. Способ определения и примеры.
69. Параллельное программирование. Понятие параллельных вычислений. Классификация параллельных вычислительных систем. Примеры параллельных вычислительных систем.
70. Параллельное программирование. Основные понятия MPI. Основные команды организации MPI-программы. Пример простой программы.
71. Параллельное программирование. Широковещательная рассылка. Операции передачи данных между двумя процессорами. Пример.
72. Параллельное программирование. Режимы передачи данных. Организация неблокирующих обменов.
73. Параллельное программирование. Коллективные операции передачи данных.

## КАРТА ЛИТЕРАТУРНОГО ОБЕСПЕЧЕНИЯ ДИСЦИПЛИНЫ

ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

Направление 44.03.05 «Педагогическое образование»

Профиль: «Физика и информатика»

Квалификация: бакалавр

**Очная** форма обучения

(общая трудоемкость 9,0 з.е.)

| № п/п | Наименование                                                                                                                           | Наличие место/ (кол-во экз.)       | Потребность | Примечания |
|-------|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|-------------|------------|
|       | <b>Обязательная литература</b>                                                                                                         |                                    |             |            |
| 1     | А.В. Могилев, Н.И. Пак, Е.К. Хеннер. Информатика. - М.: Академа, 2001.                                                                 | Отраслевая библиотека<br>ИМФИ (10) | 20          |            |
| 2     | Ю. Магда. Ассемблер для процессоров Intel Pentium. – Питер, 2006.                                                                      | Отраслевая библиотека<br>ИМФИ (10) | 20          |            |
| 3     | А. Адаменко, А. Кучуков. Логическое программирование и Visual Prolog. – СПб, БХВ-Петербург, 2003                                       | Отраслевая библиотека<br>ИМФИ (5)  | 20          |            |
| 4     | Л.В. Городняя. <a href="#">Основы функционального программирования</a><br>Интернет-университет информационных технологий - БИНОМ, 2004 | Отраслевая библиотека<br>ИМФИ (5)  | 20          |            |
| 5     | В.П. Гергель. Теория и практика параллельных вычислений. - Интернет-университет информационных технологий - БИНОМ, 2007                | Отраслевая библиотека<br>ИМФИ (5)  | 20          |            |
|       | <b>Дополнительная литература</b>                                                                                                       |                                    |             |            |
| 6     | Дональд Э. Кнут. Искусство программирования. В 3-х томах                                                                               | Отраслевая библиотека<br>ИМФИ (3)  | 5           |            |
| 7     | Роберт У. Себеста. <a href="#">Основные концепции языков программирования</a> .                                                        | Отраслевая библиотека<br>ИМФИ (7)  | 5           |            |
| 8     | Герберт Шилдт. Полный справочник по С++. Четвертое издание. – М.-СПб.-Киев: Изд. дом «Вильямс», 2009.                                  | Отраслевая библиотека<br>ИМФИ (5)  | 5           |            |



|    |                                                                                                                                                       |                                   |   |  |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|---|--|
| 9  | Системы искусственного интеллекта. Практический курс. Учебное пособие под ред. И.Ф. Астаховой. – М.: Бином, Физматлит, 2008.                          | Отраслевая библиотека<br>ИМФИ (3) | 5 |  |
| 10 | С. Немнюгин, О. Стесик. Параллельное программирование для многопроцессорных вычислительных систем. – Спб.: БХВ-Петербург, 2002.                       | Отраслевая библиотека<br>ИМФИ (2) | 5 |  |
| 11 | Тей, П. Грибомон, Ж. Луи и др. Логический подход к искусственному интеллекту: от классической логики к логическому программированию. - М.: Мир, 1990. | Отраслевая библиотека<br>ИМФИ (5) | 5 |  |
| 12 | В.В. Воеводин, Вл.В. Воеводин. Параллельные вычисления. - Спб.: БХВ-Петербург, 2004.                                                                  | Отраслевая библиотека<br>ИМФИ (3) | 5 |  |
|    | <b>Методические пособия, рекомендации</b>                                                                                                             |                                   |   |  |
| 13 | А.С. Денисюк, М.И. Ганжа «Введение в Пролог» - Красноярск, 1994.                                                                                      | Отраслевая библиотека<br>ИМФИ (2) | 2 |  |
| 14 | А.С. Антонов. Введение в параллельные вычисления. – М., 2002.                                                                                         | Отраслевая библиотека<br>ИМФИ (2) | 2 |  |

## КАРТА МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ ДИСЦИПЛИНЫ

ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

Направление 44.03.05 «Педагогическое образование»

Профиль: «Физика и информатика»

Квалификация: бакалавр

**Очная** форма обучения

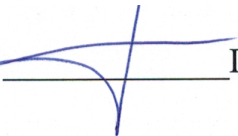
(общая трудоемкость 9,0 з.е.)

| Аудитория                                                             | Оборудование                                                                                           |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Лекционные аудитории</b>                                           |                                                                                                        |
| Ул. Перенсона ,7. ауд. № 3-02                                         | ПК с ОС Windows, проектор мультимедиа, интерактивная доска SMART-board. маркерная доска                |
| Ул. Перенсона ,7. ауд. № 2-04                                         | ПК с ОС Windows, проектор мультимедиа, интерактивная доска SMART-board. маркерная доска                |
| <b>Аудитории для практических (семинарских)/ лабораторных занятий</b> |                                                                                                        |
| Ул. Перенсона ,7. ауд. 2-04                                           | 10 ПК с ОС Windows + MS Office, проектор мультимедиа, интерактивная доска SMART-board. маркерная доска |

## ЛИСТ ВНЕСЕНИЯ ИЗМЕНЕНИЙ

Дополнения и изменения в учебной программе на 2016/2017 учебный год нет.

Рабочая программа утверждена на заседании базовой кафедры информатики и ИТ в образовании "05" октября 2016 г. (протокол заседания кафедры № 03)

Заведующий кафедрой  Пак Н.И.

Директор  Чиганов А.С.