

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
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
ИНЖЕНЕРНЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Направление подготовки/специальность:
44.04.01 Педагогическое образование

Направленность (профиль) или специализация образовательной программы:
Инженерное образование (с применением сетевой формы)
**с Сибирским федеральным университетом*

квалификация (степень):
магистр

Красноярск 2021

Рабочая программа дисциплины «Инженерные языки программирования»
составлена канд. физ.-мат. наук, доцентом кафедры ИИТвО Романовым Д.В.
(должность и ФИО преподавателя)

Рабочая программа дисциплины обсуждена на заседании кафедры Информатики и
информационных технологий в образовании (ИИТвО)

протокол № 11 от "20" мая 2020 г.

Заведующий кафедрой




(ф.и.о., подпись) Н.И. Пак

Одобрено научно-методическим советом ИМФИ

протокол № 8 от "20" мая 2020 г.

Председатель



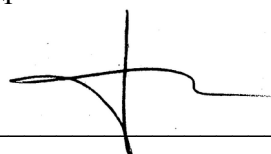
(ф.и.о., подпись) С.В. Бортновский

Рабочая программа дисциплины актуализирована
канд. физ.-мат. наук, доцентом кафедры ИИТвО Романовым Д.В.

Программа пересмотрена и одобрена на заседании кафедры

"12" мая 2021 г., протокол № 9

Заведующий кафедрой



Пак Н.И

Одобрено НМСС(Н)

"21" мая 2021 г., протокол №7

Председатель



Бортновский С.В.

Содержание

1. Пояснительная записка.....	4
1.1 Место дисциплины в структуре образовательной программы.....	4
1.2 Общая трудоёмкость дисциплины.....	4
1.3 Цели освоения дисциплины.....	4
1.4 Планируемые результаты обучения.....	5
1.5 Контроль результатов освоения дисциплины.....	7
1.6 Используемые образовательные технологии.....	7
2. Организационно-методические документы.....	9
2.1. Технологическая карта обучения дисциплине.....	9
2.2 Содержание основных разделов и тем дисциплины.....	12
2.3. Методические рекомендации по освоению дисциплины для обучающихся образовательной программы.....	13
2.4. Технологическая карта рейтинга дисциплины.....	17
3. Фонд оценочных средств (контрольно-измерительные материалы).....	19
3.1. Назначение фонда оценочных средств.....	20
3.2. Оценочные средства.....	22
3.3. Фонд оценочных средств для промежуточной аттестации.....	24
4. Лист внесения изменений.....	39
4.1. Дополнения и изменения в рабочей программе дисциплины на 2020/2021 учебный год.....	39
4.1. Дополнения и изменения в рабочей программе дисциплины на 2021/2022 учебный год.....	40
5. Карта литературного обеспечения дисциплины (включая электронные ресурсы)	41
6. Карта материально-технической базы дисциплины.....	44

1. Пояснительная записка

1.1 Место дисциплины в структуре образовательной программы

Программа дисциплины «Инженерные языки программирования» отвечает требованиям федерального государственного образовательного стандарта высшего образования (далее – ФГОС ВО) по направлению подготовки 44.04.01 Педагогическое образование (уровень магистратуры), утвержденного приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. N 126 и профессионального стандарта «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)», утвержденного приказом Министерства труда и социальной защиты РФ от 18 октября 2013 г. №544н.

Дисциплина относится к дисциплинам вариативной части учебного плана основной образовательной программы, изучается во втором семестре, индекс дисциплины в учебном плане Б1.ВДП.01.01.

1.2 Общая трудоёмкость дисциплины

Общая трудоёмкость дисциплины «Инженерные языки программирования» составляет 108 часов (3 ЗЕТ), из них:

Контактных (аудиторных) часов 34:

Лекций – 6

Лабораторных работ – 18

Практических работ – 10

Часов самостоятельной работы – 74

1.3 Цели освоения дисциплины

Программирование и алгоритмическое мышление являются одними из базовых инструментов в арсенале инженера, позволяющими как выполнять задачи исследования и моделирования с помощью специальных сред (LabView, Matlab, ANSYS, Maple, Wolfram Alpha и др.), так и расширять их возможности, поэтому учебный курс, посвящённый ключевым языкам индустрии и их фундаментальным концепциям, занимает важное место в системе подготовки магистра и имеет как мировоззренческое, так и прикладное значение.

Ключевые цели курса — освоение техники управления сложностью при работе с алгоритмами, формирование системы понятий, знаний, умений и навыков работы в

структурной, объектно-ориентированной и функциональной парадигмах при работе на языках С, С++ и Python. При этом изучаются методы проектирования, анализа и создания программного обеспечения, проводится знакомство со множеством открытых библиотек для решения прикладных задач. Как результат, формируется ряд ключевых компетенций, а также актуализируются и закрепляются алгоритмическое и вычислительное мышления.

Выбор цели обусловлен следующими положениями: во-первых, программирование является существенной частью подготовки при решении будущими педагогами и инженерами задач исследования и моделирования. Во-вторых, программирование является предметом, развивающим способность к алгоритмической мыслительной деятельности и создающим фундамент для реализации множества межпредметных связей. В-третьих, умение программировать на языке Питон является базовым требованием во множестве сфер деятельности специалиста как открывающее доступ ко множеству мощных бесплатных инструментов автоматизации и выполнения крайне широкого круга задач. В-четвёртых, многие среды для профессиональной работы математиков (Maple, Matlab, Wolfram Mathematica, Maxima и другие) опираются на вычислительное мышление, формируемое в ходе изучения предмета.

1.4 Планируемые результаты обучения.

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ПК-1 — способен реализовывать образовательные программы в соответствии с требованиями федеральных государственных образовательных стандартов.

ПК-3 — способен организовывать научно-исследовательскую деятельность обучающихся.

ПК-5 — готов к организационно-методическому сопровождению команд обучающихся для участия в олимпиадах и конкурсах инженерно-технологической направленности.

Планируемые результаты обучения

Задачи освоения дисциплины	Планируемые результаты обучения по дисциплине (дескрипторы)	Код результата обучения (компетенция)
Формирование культуры работы в динамичной, быст-	Сформировать представление о языке как: <ul style="list-style-type: none"> • знаково-символьной системе записи и анализа алгоритмов; 	ПК-1 ПК-3 ПК-5

<p>развивающейся области, регулируемой сообществом создателей и пользователей продукта в масштабах планеты.</p>	<ul style="list-style-type: none"> • носители информации о технике выполнения задачи в форме, облегчающей использование одной или нескольких парадигм разработки прикладного ПО; • кризисах, возникающих при росте сложности разрабатываемого ПО, методиках их локализации и разрешения; • объекте, описываемом промышленным стандартом, и о процессах разработки и внедрения стандартов. 	
<p>Знакомство и получение практического опыта проектирования и разработки комплексного ПО.</p>	<p>Выпускник владеет:</p> <ul style="list-style-type: none"> • знаниями структурного, объектно-ориентированного и функционального программирования; • альтернативными и смежными методами разработки (автоматное, логическое программирование); • основами современных математических и логико-семантических подходов в постановке формализации, формулировании и реализации задач программирования; <p>а также</p> <ul style="list-style-type: none"> • имеет чёткое представление об основных методах и парадигмах решения программных задач; • способен использовать математический аппарат, методологию программирования и современные компьютерные технологии. 	<p>ПК-1 ПК-3 ПК-5</p>
<p>Базовое обучение классических языкам программирования и методикам разработки ПО.</p>	<p>Студент должен знать:</p> <ul style="list-style-type: none"> • сущность структурного, ОО и функционального подходов к программированию, их особенности и реализации с помощью языков С, С++, Python; • простые и сложные типы данных и их применение при написании программ; • назначение и состав базовых библиотек; • создание и назначение динамических структур данных. <p>уметь:</p> <ul style="list-style-type: none"> • разрабатывать линейные, разветвляющиеся и циклические алгоритмы решения задач; • самостоятельно программировать основные компоненты базовых алгоритмов; • использовать специализированное программное обеспечение для программирования; • разрабатывать алгоритмы, использующие возможности языка. <p>владеть навыками:</p> <ul style="list-style-type: none"> • разработки алгоритмов в различных парадигмах; • программирования сложных алгоритмов с использованием динамических структур. 	<p>ПК-1 ПК-3 ПК-5</p>

1.5 Контроль результатов освоения дисциплины

Методы текущего контроля: выполнение практических и теоретических заданий к каждому занятию (аналитический обзор исследований, изучение и обобщение методического опыта, работа над проектным заданием, решение проблемных ситуаций), посещение лекций, практических занятий, презентация результатов текущей работы.

Методы промежуточного контроля: входное тестирование, тематический кейс, проектное задание.

Итоговый контроль: защита проектного задания.

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

1.6 Используемые образовательные технологии

1. Лекции, семинары.
2. Перевернутый класс.
3. Проблемное обучение.
4. Эвристическая беседа.
5. Работа на тренажёре.

Дисциплина подразумевает интенсивную индивидуальную практическую работу с интерактивной обратной связью благодаря опоре на среды с автоматическим тестированием программы с предоставлением ожидаемых и полученных результатов, вследствие чего используются такие технологии, как перевернутый класс, проблемное обучение, технология мастерских, работа в малых группах, наставничество.

В связи с практической направленностью и инженерно-прикладным характером изучаемого материала, на лекциях значительную долю контактной работы с обучающимися занимают мозговые штурмы, дискуссии, эвристические беседы, разбор практико-ориентированных заданий.

Активно используются средства информационных технологий для проведения интерактивных исследований фрагментов кода и алгоритмов в аудитории, визуального представления состояния исполнителя, сетевого сопровождения и контроля самостоятельной работы через систему обмена сообщениями.

Автор благодарит разработчиков сред pythontutor.org и pythontutor.ru за предоставленные уникальные дидактические и методические возможности.

2. Организационно-методические документы

2.1. Технологическая карта обучения дисциплине

Наименование дисциплины: Инженерные языки программирования

Для обучающихся образовательной программы: 44.04.01 Педагогическое образование

Направленность (профиль) или специализация образовательной программы: Инженерное образование
(с применением сетевой формы) *с Сибирским федеральным университетом

Форма обучения: очная

Общая трудоемкость дисциплины: 3 з.е.

Наименование разделов и тем дисциплины	Всего часов	Аудиторных часов				Внеауди- торных часов	Формы и методы контроля
		Всего	Лекций	Семинаров	Практик		
Проектирование и разработка ПО на языках Си и Си++							
Понятие исполнителя. Архитектура фон Неймана. Представления кода и данных. Run-time и compile-time исполнение. Процедурное программирование.	6	2	2			4	Квиз
Знакомство с IDE. Ввод и вывод. Простые программы. if, while, for. int, double, char.	6	2		2		4	Лабораторная работа
Массивы и структуры. Алгоритмы обработки: флаги, аккумуляторы; сортировка.	6	2		2		4	Лабораторная работа
Структуры и функции. Задачи на работу с базой данных хранимой в массиве структур.	7	2		2		5	Лабораторная работа
Функции и структуры как API. Управление сложностью. Процедурная парадигма. Примеры. Понятие ссылки и указателя. Анализ реальных проектов.	6	2	2			4	Эвристическая беседа

Алгоритмическая сложность. Задача хранения пар ключ-значение. Хэш-карта.	6	2		2		4	Тест
Проблемы структурного программирования. ООП. RAII. Compile-time инварианты и гарантии. Простой конструктор. Метод.	6	2		2		4	Контрольная работа
Защита данных. Инкапсуляция. Полиморфизм. Система типов, анализ.	7	2		2		5	Эвристическая беседа
ООП парадигма как DSL. STL: vector, map, string, map. Задачи на графах.	7	2		2		5	Лабораторная работа
Проектирование и разработка ПО на языке Питон							
Язык Питон. IDE. REPL. Система хранения данных. Типы. Инструменты интроспекции, dir, if, while, for, range.	6	2	2			4	Лабораторная работа
list comprehension. zip. Срезы. Функции. Именованные и позиционные аргументы. Функция как интерфейс и как API.	6	2		2		4	Лабораторная работа
Решение ранее пройденных задач на Питон, анализ возможностей языка. Анализ проектировочных решений при создании Си, Си++, Питон.	6	2		2		4	Лабораторная работа
map-reduce. Лямбда-функции. Функциональное программирование.	4	2			2	2	Лабораторная работа
API сторонних сервисов. Яндекс-службы. request, pip, PIL. PyPI.	5	2			2	3	Лабораторная работа
Проект							
Выполнение проекта.	22	4			4	18	Отчёт
Защита проекта.	2	2			2		Презентация и защита проекта
ИТОГО	108	34	6	18	10	74	

Форма итогового контроля по учебному плану							Нет
---	--	--	--	--	--	--	-----

2.2 Содержание основных разделов и тем дисциплины

Модуль 1: Проектирование и разработка ПО на языках Си и Си++

Тема 1.1. Ключевые сведения об архитектуре ПК. Понятие исполнителя, архитектура фон Неймана, система команд, состояние исполнителя. АЛУ, УУ, память, адресация. Низкоуровневые языки. Ассемблер. Процедурное программирование. Run-time и compile-time исполнение.

Тема 1.2. Структурное программирование. Сущность и свойства структурного программирования. Блок-схемы основных алгоритмических конструкций. Среда Code Block.

Тема 1.3. Массивы и структуры. Алгоритмы обработки: флаги, аккумуляторы; сортировка.

Тема 1.4. Структуры и функции. Задачи на работу с базой данных хранимой в массиве структур.

Модуль 2: Алгоритмы и структуры данных

Тема 2.1. Функции и структуры как API. Управление сложностью. Процедурная парадигма. Примеры. Понятие ссылки и указателя. Анализ реальных проектов (код браузера, id3-id5 tech engines), и выделение соответствующих иерархий абстракций.

Тема 2.2. Алгоритмическая сложность. Задача хранения пар ключ-значение. Хэш-карта.

Модуль 3: ООП

Тема 3.1. Проблемы структурного программирования. ООП. RAII. Compile-time инварианты и гарантии. Простой конструктор. Метод.

Тема 3.2. Защита данных. Инкапсуляция. Полиморфизм. Система типов, анализ.

Тема 3.3. ООП парадигма как DSL. STL: vector, map, string, map. Задачи на графах.

Модуль 4: Проектирование и разработка ПО на языке Питон

Тема 4.1. Язык Питон. IDE.REPL. Система хранения данных. Типы. Инструменты интроспекции, dir, if, while, for, range.

Тема 4.2. list comprehension. Срезы, zip. Функции. Именованные и позиционные аргументы. Функция как интерфейс и как API.

Тема 4.3. Решение ранее пройденных задач на Питон, анализ возможностей языка. Анализ проектировочных решений при создании Си, Си++, Питон.

Тема 4.4. map-reduce. Лямбда-функции. Функциональное программирование.

Тема 4.5. API сторонних сервисов. Яндекс-службы. request, pip, PIL. PyPI.

Модуль 5: проект

Выполнение и защита проекта.

2.3. Методические рекомендации по освоению дисциплины для обучающихся образовательной программы

Введение

Методические рекомендации содержат:

- Рекомендации по организации работы студента на лекциях и практических занятиях.
- Рекомендации по организации самостоятельной работы студента.
- Рекомендации по работе в модульно-рейтинговой системе.
- Советы по подготовке к экзамену.

Методические рекомендации по работе 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. В этом помогает конспектирование сути материала, излагаемого преподавателем (Во время конспектирования в работу включаются зрительная, аудиальная и моторно-двигательная память, позволяющие эффективно усвоить лекционный материал.) Главное, что конспектирование лекции – это не диктант. Для успешной работы студент только выделяет суть, и фиксирует её «своими словами» в объёме, достаточном для гарантированного воспроизведения. Это гораздо более эффективно, чем запись «под диктовку». В ходе возникновения трудностей следует относиться к этому как к признаку правильного хода работы, чётко сформулировать непонимаемый фрагмент высказывания лектора и задать вопрос, стараясь не нарушать ритм и ход лекции (для вопросов в лекции специально оставлены окна в несколько минут). Часто это помогает всем студентам лучше осознать материал.

Следует быть готовым к тому, что на лекциях периодически проводится письменный опрос студентов по материалам лекций (квиз). 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, повысить уровень дисциплины в ходе совместной очной работы.

Программирование — практическая дисциплина, многие техники которой были созданы для решения целых классов трудностей, стоящих перед человеком. Осознание этих трудностей и спектра возможностей их преодоления намного полезнее заучивания конкретных решений — помните это.

Методические рекомендации по работе на практических занятиях

Наряду с прослушиванием лекций по курсу, ключевое место в учебном процессе занимают лабораторные занятия для апробации, закрепления и переосмысления полученных студентами знаний, содержащих большую долю практического и прикладного характера.

Перед практическим занятием студенту необходимо освежить в памяти теоретический материал по теме практического занятия. Для этого следует обратиться к соответствующим главам учебника, конспекту лекций.

Каждое занятие начинается с повторения необходимых элементов теоретического материала по соответствующей теме. Для самопроверки, студенты должны уметь чётко ответить на вопросы, поставленные преподавателем. По характеру ответов преподаватель делает вывод о том, насколько тот или иной студент готов к выполнению упражнений.

После такой проверки студентам предлагается выполнить соответствующие задания и варианты задачи. 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 советы.

По истечении времени, необходимого для решения задачи, один из студентов может быть вызван для её выполнения на доске.

В конце занятия преподаватель подводит его итоги, даёт оценку активности студентов и уровня их знаний, вносит баллы в рейтинговую таблицу.

Каждому студенту необходимо основательно закреплять полученные знания и выработать навыки самостоятельной научной работы. С этой целью в течение семестра студент должен выполнить домашние работы. Часть лабораторных допускается выполнять дома, особенно при опережении графика сдачи, поскольку в процессе сдачи авторство и глубина понимания материала крайне легко проверяется индивидуальными вопросами, к чему тоже следует быть готовым.

Методические рекомендации по самостоятельной работе 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 семинарах, но дома в ходе самостоятельной работы.

Внеаудиторная самостоятельная работа включает выполнение заданий по каждому разделу курса, многие из которых доступны в сети Интернет и снабжены системами автоматической проверки корректности.

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 модулям (100) отвечает 100%-ному усвоению материала. Минимальное количество баллов в каждом модуле является обязательным и не может быть заменено набором баллов в других модулях, за исключением ситуации, когда минимальное количество баллов по модулю определено как нулевое.

Дисциплинарный модуль считается изученным, если студент набрал количество баллов в установленном диапазоне. Соответствие процентов рейтинга академической оценке даёт таблица ниже.

Преподаватель имеет право по своему усмотрению добавлять студенту определённое 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 приёму 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.

2.4. Технологическая карта рейтинга дисциплины

ВХОДНОЙ КОНТРОЛЬ			
(проверка «остаточных» знаний по ранее изученным смежным дисциплинам)			
	Форма работы*	Количество баллов 5 %	
		min	max
	Тестирование	0	5
Итого		0	5

БАЗОВЫЙ РАЗДЕЛ № 1			
	Форма работы*	Количество баллов 35 %	
		min	max
Текущая работа	Лабораторная работа №1	3	6
	Лабораторная работа №2	3	6
	Лабораторная работа №3	3	6
	Тест	2	4
Промежуточный рейтинг-контроль	Лабораторная работа №4	3	6
	Контрольная работа	2	7
Итого		16	35

БАЗОВЫЙ РАЗДЕЛ № 2			
	Форма работы*	Количество баллов 30 %	
		min	max
Текущая работа	Лабораторная работа №5	3	6
	Лабораторная работа №6	3	6
	Лабораторная работа №7	3	6
	Лабораторная работа №8	3	6
	Лабораторная работа №9	3	6
Итого		15	30

ЗАЩИТА ПРОЕКТА			
----------------	--	--	--

Содержание	Форма работы*	Количество баллов 30 %	
		min	max
	Тестирование/ экзамен	15	30
Итого		15	30

*Перечень форм работы текущей аттестации определяется кафедрой или ведущим преподавателем

Соответствие рейтинговых баллов и академической оценки:

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

*При количестве рейтинговых баллов более 100, необходимо рассчитывать рейтинг учебных достижений студента для определения оценки кратно 100 баллов.

3. 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 № 9
от «12» мая 2021 г.



Пак Н.И.

ОДОБРЕНО
на заседании научно-методического совета
направления подготовки
Протокол № 7 от «21» мая 2021 г.



Бортновский С.В.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля и промежуточной аттестации обучающихся

«Инженерные языки программирования»

(наименование дисциплины/модуля/вида практики)

Направление подготовки: *44.04.01 Педагогическое образование*

направленность (профиль) образовательной программы *Инженерное образование*

*(с применением сетевой формы) *с Сибирским федеральным университетом*

по очной форме обучения (общая трудоемкость дисциплины 3 з.е.)

Составители:

канд. физ.-мат. наук, доцент кафедры ИИТвО Романов Д. В.

канд. пед. наук, доцент кафедры ИИТвО Сокольская М.А.

3.1. Назначение фонда оценочных средств

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

ФОС по дисциплине решает **задачи**:

1. Управление процессом приобретения обучающимися необходимых знаний, умений, навыков и формирования компетенций, определённых 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.

ФОС разработан на основании нормативных документов:

- ФГОС высшего образования по направлению подготовки 44.04.01 Педагогическое образование, квалификация (степень) «Магистр»
- образовательной программы высшего образования по направлению подготовки 44.04.01 Педагогическое образование, квалификация (степень) «Магистр»
- 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.

Перечень компетенций с указанием этапов их формирования в процессе изучения:

ПК-1 — способен реализовывать образовательные программы в соответствии с требованиями федеральных государственных образовательных стандартов.

ПК-3 — способен организовывать научно-исследовательскую деятельность обучающихся.

ПК-5 — готов к организационно-методическому сопровождению команд обучающихся для участия в олимпиадах и конкурсах инженерно-технологической направленности.

3.2. Оценочные средства

Компетенция	Дисциплины, практики, участвующие в формировании данной компетенции	Тип контроля	Оценочное средство/КИМ	
			№	Форма
ПК-1	Модуль 1 "Методология исследования в образовании": Методология и методы научного педагогического исследования. Современные подходы в научных педагогических исследованиях. Учебная практика: научно-исследовательская работа. Модуль 3 "Основы организации профессиональной педагогической деятельности": Информационно-коммуникационные технологии в профессиональной деятельности. Мониторинг образовательных результатов. Модуль 4 "Образовательная робототехника": Инженерные языки программирования. Практикум по робототехнике. Организация современной инженерной лаборатории. Учебная практика: ознакомительная практика. Модуль 5 "Предметно-технологический": Методика STEM-обучение. Организация исследовательских проектов обучающихся с использованием ресурсов технопарков. Методика формирования метапредметных результатов в общеобразовательной школе. Модуль 5 "Инженерное проектирование": Жизненный цикл инженерного проекта. Современные промышленные технологии. Методика сопровождения исследовательской деятельности обучающихся с использованием ресурсов технопарков. Модуль 6 "Проектирование креативно-ориентированной образовательной среды для классов инженерно-технологической направленности": Педагогические технологии смешанного обучения предмету физико-математического цикла. Психология и педагогика профессионального самоопределения. Проектирование дополнительных образовательных программ для особо мотивированных обучающихся. Учебная практика. Ознакомительная практика. Производственная практика. Педагогическая практика. Преддипломная практика. Подготовка к сдаче и сдача государственного экзамена. Выполнение и защита выпускной квалификационной работы.	Текущий контроль успеваемости Промежуточная аттестация	1-3	Выполнение ЛР Защита проекта
ПК-3	Модуль 1 "Методология исследования в образовании": Современные проблемы науки и образования. Учебная практика: научно-исследовательская работа. Модуль 2 "Педагогическое проектирование": Теоретические основы педагогического проектирования. Проектирование образовательных программ. Проектирование систем исследовательской работы обучающихся. Модуль 3 "Основы организации	Текущий контроль успеваемости	1-3	Выполнение ЛР Защита проекта

	<p>профессиональной педагогической деятельности": Деловой иностранный язык. Модуль 4 "Образовательная робототехника": Инженерные языки программирования. Практикум по робототехнике. Организация современной инженерной лаборатории. Учебная практика: ознакомительная практика. Элективные модули. Модуль 5 "Инженерное проектирование": Жизненный цикл инженерного проекта. Современные промышленные технологии. Методика сопровождения исследовательской деятельности обучающихся с использованием ресурсов технопарков. Модуль 6 "Проектирование креативно-ориентированной образовательной среды для классов инженерно-технологической направленности": Педагогические технологии смешанного обучения предмету физико-математического цикла. Психология и педагогика профессионального самоопределения. Проектирование дополнительных образовательных программ для особо мотивированных обучающихся. Учебная практика. Ознакомительная практика. Производственная практика. Научно-исследовательская работа. Преддипломная практика. Подготовка к сдаче и сдача государственного экзамена. Выполнение и защита выпускной квалификационной работы.</p>	Промежуточная аттестация		
ПК-5	<p>Модуль 4 "Образовательная робототехника": Инженерные языки программирования. Практикум по робототехнике. Организация современной инженерной лаборатории. Учебная практика: ознакомительная практика. Элективные модули. Модуль 5 "Предметно-технологический": Организация исследовательских проектов обучающихся с использованием ресурсов технопарков. Методика формирования метапредметных результатов в общеобразовательной школе. Модуль 5 "Инженерное проектирование": Жизненный цикл инженерного проекта. Современные промышленные технологии. Методика сопровождения исследовательской деятельности обучающихся с использованием ресурсов технопарков. Модуль 6 "Проектирование креативно-ориентированной образовательной среды для классов инженерно-технологической направленности": Педагогические технологии смешанного обучения предмету физико-математического цикла. Психология и педагогика профессионального самоопределения. Производственная практика. Научно-исследовательская работа. Выполнение и защита выпускной квалификационной работы.</p>	Текущий контроль успеваемости и Промежуточная аттестация	1-3	Выполнение ЛР Защита проекта

3.3. Фонд оценочных средств для промежуточной аттестации

Фонды оценочных средств включают:

3.3.1. Вопросы и задания к промежуточной аттестации

3.3.2. Темы для эвристических бесед и рефератов

3.3.3. Критерии оценивания по оценочным средствам 3.3.1, 3.3.2

3.3.1. Оценочное средство 1 «Вопросы и задания к промежуточной аттестации»

- 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.
 - e) Python.
 - f) Pascal.
- 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 выполниться 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) aaacc
- e) bbccc
- 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) ббб
- 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?

- a) $0 \text{xFF} \& 5 \gg 1 + 1$
- b) 1
- c) 2
- d) 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

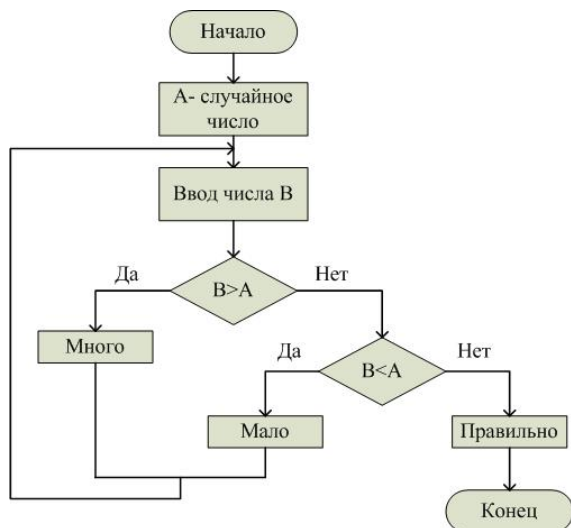
- a) Первый элемент списка;
- b) Сумма элементов списка;
- c) Сумма первого и последнего элементов списка;
- d) Количество элементов списка;
- e) Количество звеньев списка, где указатель на следующее звено не nil.

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?

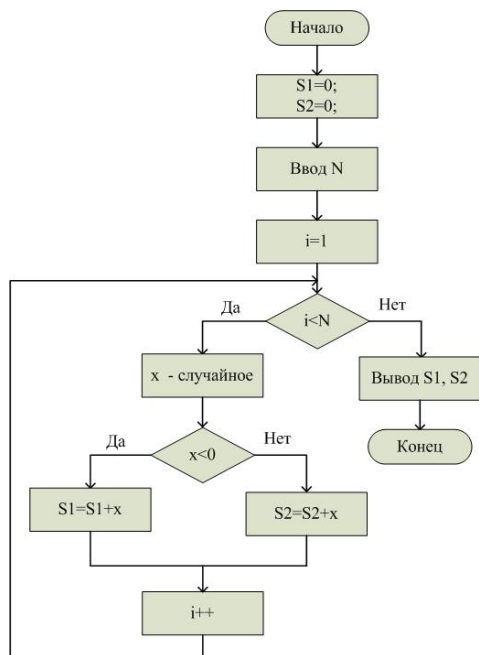
43) Даны две дроби A/B и C/D (A, B, C, D — натуральные числа). Составить программу для деления дроби на дробь. Результат должен быть несократимой дробью. Какую часть в

предложенной задаче вы можете выделить как подпрограмму?

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



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



3.3.2. Темы для эвристических бесед и рефератов

- 1) Принципы структурного программирования. Основные алгоритмические конструкции.
- 2) Язык C/C++/Питон. Структура программы. Типы данных.
- 3) Язык C/C++/Питон. Библиотеки функций. Ввод-вывод данных. Примеры.
- 4) Язык C/C++/Питон. Организация ветвлений (if, :?). Конструкция множественного выбора (switch). Пример.
- 5) Язык C/C++/Питон. Организация циклов с постусловием и предусловием. Примеры.
- 6) Язык C/C++/Питон. Цикл с параметром. Его особенности и возможности. Примеры.
- 7) Язык C/C++/Питон. Указатели. Операторы работы с указателями. Арифметика указателей. Примеры.
- 8) Язык C/C++/Питон. Пользовательские функции. Способы передачи данных в функцию. Примеры.
- 9) Язык C/C++/Питон. Статические одномерные массивы. Создание массивов и обработка элементов массивов.
- 10) Передача массивов в функцию. Примеры.
- 11) Язык C/C++. Статические двумерные массивы. Создание массивов и обработка элементов массивов.
- 12) Передача массивов в функцию. Примеры.
- 13) Язык C/C++/Питон. Динамические одномерные массивы. Создание одномерных динамических массивов и обработка элементов массивов. Передача массивов в функцию. Связь массивов и указателей. Примеры
- 14) Язык C/C++/Питон. Динамические двумерные массивы. Создание и удаление двумерных динамических массивов и обработка элементов массивов. Передача массивов в функцию. Связь массивов и указателей. Примеры
- 15) Язык C/C++/Питон. Строки. Обработка строк. Связь строк, массивов и указателей. Пример.
- 16) Язык C/C++/Питон. Структуры. Описание структуры в программе. Массивы структур. Пример.
- 17) Язык C/C++/Питон. Динамические односвязные списки. Примеры процедур обработки списков.

- 18) Язык C/C++/Питон. Динамические списки. Стеки. Примеры процедур обработки стека.
- 19) Язык C/C++/Питон. Динамические списки. Очереди. Примеры процедур обработки очереди.
- 20) Язык C/C++/Питон. Алгоритмы поиска и сортировки
- 21) Основы анализа эффективности алгоритмов. Трудоёмкость алгоритмов. Временная сложность алгоритмов.
- 22) Классы сложности алгоритмов.
- 23) Основы анализа эффективности алгоритмов. Методы оценки ресурсной эффективности алгоритмов.
- 24) Метод получения функции трудоёмкости. Пример.
- 25) Функциональное программирование. Принципы функционального программирования. Примеры.
- 26) Язык Питон. Основные типы данных. Достоинства и недостатки. Интроспекция. Примеры.
- 27) Язык Питон. Тип данных dict, его возможности. Пример.
- 28) Язык Питон. Тип данных set, его возможности. Пример.
- 29) Язык Питон и функциональное программирование. Декораторы. map. reduce. Примеры.

3.3.3. Критерии оценивания по оценочным средствам 3.3.1, 3.3.2

Формируемые компетенции	Продвинутый уровень сформированности компетенций	Базовый уровень сформированности компетенций	Пороговый уровень сформированности компетенций
	(20 - 23 балла) отлично	(16 - 19 баллов) хорошо	(13 - 15 баллов)* Удовлетворительно
ПК-1 — способен реализовывать образовательные программы в соответствии с требованиями федеральных государственных образовательных стандартов.	Имеет полное представление о теоретическом и практическом аппарате дисциплины; способен доработать его самостоятельно с предварительным выделением собственных дефицитов.	Демонстрирует хороший уровень готовности реализовывать образовательные программы по информатике в соответствии с требованиями ФГОС ООО и ФГОС СПОО.	Обучающийся демонстрирует достаточный уровень готовности реализовывать образовательные программы по информатике в соответствии с требованиями ФГОС

	Способен проектировать примеры и задачи любого уровня сложности.	Имеет опыт работы с соответствующими нормативными документами.	ООО и ФГОС СПОО
ПК-3 — способен организовывать научно-исследовательскую деятельность обучающихся.	Дополнительно: способен быстро и качественно реферировать поступающую литературу, анализировать дайджесты, создавать прототипы для иллюстрации своих положений. Имеет практический опыт решения реальных задач производства или участия в хакатонах и митапах, имеет портфолио готовых проектов и работ.	Дополнительно: имеет представление о смежных дисциплинах, использующих изучаемый аппарат языков и сопутствующих библиотек, знает и начал изучать доступные онлайн-курсы по соответствующей тематике, ознакомлен с литературой. Способен создавать модели для иллюстрации положений, умеет вести дискуссию, формулировать и защищать свои положения.	Имеет опыт научно-исследовательской работы с обучающимися, получил соответствующую методическую и предметную подготовку, владеет методами и знает методологию научно-исследовательской деятельности.
ПК-5 — готов к организационно-методическому сопровождению команд обучающихся для участия в олимпиадах и конкурсах инженерно-технологической направленности.	Обучающийся демонстрирует владение всеми изученными парадигмами программирования, истории и причин их создания, классах решаемых проблем, приводит примеры изученных материалов научных исследований и ссылается на личный опыт. Способен продемонстрировать любое изучаемое положение самостоятельно для любого из предложенных кейсов.	Обучающийся демонстрирует понимание изученных методов получения научного и прикладного знания в области разработки ПО, приводит примеры изученных материалов. Обучающийся демонстрирует владение полученными знаниями при выборе адекватных структур данных и кейсов для демонстрации преподаваемого материала, способен объяснить выбор.	Обучающийся демонстрирует знание принципов разработки ПО, место и роль в современном научном исследовании и производстве; способен провести анализ произвольного кейса с опорой на сеть интернет, анализируя предложенные в сети решения на предмет правдоподобия и наличия внутренних противоречий и/или противоречий с ранее изученным материалом.

*Менее 13 баллов – компетенция не сформирована

4. Лист внесения изменений

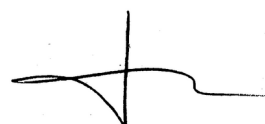
4.1. Дополнения и изменения в рабочей программе дисциплины на 2020/2021 учебный год

Рабочая программа дисциплины разработана впервые для данной ОПОП.

Рабочая программа дисциплины обсуждена на заседании кафедры Информатики и информационных технологий в образовании (ИИТвО)

протокол № 11 от "20" мая 2020 г.

Заведующий кафедрой



Н.И. Пак

(ф.и.о., подпись)

Одобрено научно-методическим советом ИМФИ

протокол № 8
от "20" мая 2020 г.

Председатель



С.В. Бортновский

(ф.и.о., подпись)

4.1. Дополнения и изменения в рабочей программе дисциплины на 2021/2022 учебный год

В программу вносятся следующие изменения:

1. Обновлены титульные листы рабочей программы, фонда оценочных средств.

Программа пересмотрена и одобрена на заседании кафедры
"12" мая 2021 г., протокол № 9

Внесенные изменения
кафедрой



Пак Н.И.

утверждаю: Заведующий

Одобрено НМСС(Н)
"21" мая 2021 г., протокол №7

Председатель



Бортновский С.В.

5. Карта литературного обеспечения дисциплины (включая электронные ресурсы)

Наименование дисциплины: Инженерные языки программирования

Для обучающихся образовательной программы: 44.04.01 Педагогическое образование

Направленность (профиль) или специализация образовательной программы: Инженерное образование
(с применением сетевой формы) *с Сибирским федеральным университетом

Форма обучения: очная

Общая трудоемкость дисциплины: 3 з.е.

Наименование	Место хранения/ электронный адрес	Кол-во экземпляров/ точек доступа
Основная литература		
Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика [Текст] : учебное пособие / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. - 4-е изд., стереотип. - М. : Академия, 2007. - 848 с. - ISBN 5-7695-3244-0	Научная библиотека КГПУ им. В.П. Астафьева	185
Ефимова, Е.А. Основы программирования на языке Visual Prolog / Е.А. Ефимова. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 266 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428996	ЭБС «Университетская библиотека онлайн»	Индивидуальный неограниченный доступ
Павловская Т.А. С/С++. Программирование на языке высокого уровня [Текст] : учебное пособие / Т.А. Павловская. - СПб. : Питер, 2002. - 464 с. - ISBN 5-318-00001-0	Научная библиотека КГПУ им. В.П. Астафьева	8

Дополнительная литература		
Зеленин В.М., Пак Н.И., Якушевич В.И. Системы искусственного интеллекта и язык Пролог в обучении [Текст]: учебное пособие и автоматизированный учебный курс для ст-в педвузов, учителей и уч-ся ст. классов / В. М. Зеленин, Н. И. Пак, В. И. Якушевич. - Красноярск : РИО КГПУ, 1999. - 200 с.	Научная библиотека КГПУ им. В.П. Астафьева	41
Денисюк А.С., Ганжа М.И. Введение в пролог [Текст]: методические указания для студентов физ.-мат. фак. / Сост. А. С. Денисюк, М. И. Ганжа. - Красноярск : КГПУ, 1994. - 48 с.	Научная библиотека КГПУ им. В.П. Астафьева	5
Николаев, Е.И. Параллельные вычисления : учебное пособие / Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2016. - 185 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=459124	ЭБС «Университетская библиотека онлайн»	Индивидуальный неограниченный доступ
Сорокин, А.А. Объектно-ориентированное программирование : учебное пособие (курс лекций) / А.А. Сорокин ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2014. - 174 с. : ил. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=457696	ЭБС «Университетская библиотека онлайн»	Индивидуальный неограниченный доступ
Ресурсы сети интернет		
Павленко В., Соломатин В., Кириенко Д. П. Pythontutor: Интерактивный учебник языка Питон [Электронный ресурс]. – URL: http://pythontutor.ru/	http://pythontutor.ru/	Свободный доступ
Philip Guo. VISUALIZE CODE AND GET LIVE HELP: Learn Python, Java, C, C++, JavaScript, and Ruby [Электронный ресурс]. – URL: http://pythontutor.com/	http://pythontutor.com/	Свободный доступ
Столяров А.В. Введение в язык Си++: [учебное пособие] / А. В. Столяров. - 4-е изд., испр. и доп.. - Москва: МАКС Пресс. - 2018. - 136с.: ил. - На яз. - . - ISBN 978-5-317-05781-7. - Тираж 150экз.	http://www.stolyarov.info/books/cppintro	Свободный доступ

Столяров А.В. Программирование: введение в профессию. 1: Азы Программирования. / А. В. Столяров. - М.: Макс Пресс, 2016. - 464 с.	http://www.stolyarov.info/books/programming_intro/vol1	Свободный доступ
Поляков К.В. Язык программирования Си / К.В. Поляков [Электронный ресурс]. – URL: http://kpolyakov.spb.ru/school/c.htm	http://kpolyakov.spb.ru/school/c.htm	Свободный доступ
Информационные справочные системы и профессиональные базы данных		
Гарант [Электронный ресурс]: информационно-правовое обеспечение : справочная правовая система. – Москва, 1992– .	Научная библиотека	локальная сеть вуза
Elibrary.ru [Электронный ресурс] : электронная библиотечная система : база данных содержит сведения об отечественных книгах и периодических изданиях по науке, технологии, медицине и образованию / Рос. информ. портал. – Москва, 2000– . – Режим доступа: http://elibrary.ru .	http://elibrary.ru	Свободный доступ
East View : универсальные базы данных [Электронный ресурс] : периодика России, Украины и стран СНГ . – Электрон.дан. – ООО ИВИС. – 2011 - .	https://dlib.eastview.com/	Индивидуальный неограниченный доступ
Межвузовская электронная библиотека (МЭБ)	https://icdlib.nspu.ru/	Индивидуальный неограниченный доступ

Согласовано:

_____ /  _____
 (должность структурного подразделения) (подпись) (Фамилия И.О.)

6. Карта материально-технической базы дисциплины

Наименование дисциплины: Инженерные языки программирования

Для обучающихся образовательной программы: 44.04.01 Педагогическое образование

Направленность (профиль) или специализация образовательной программы:

Инженерное образование

(с применением сетевой формы) *с Сибирским федеральным университетом

Форма обучения: очная

Общая трудоемкость дисциплины: 3 з.е.

Аудитория	Оборудование
Лекционные аудитории	
Ул. Перенсона, 7 ауд. № 3-02	<p>Оборудование: Компьютер- 1шт., интерактивная доска - 1 шт., система видеоконференцсвязи Policom – 1 шт. (без сети), учебная доска-1шт.</p> <p>Программное обеспечение Linux Mint – (Свободная лицензия GPL)</p>
Ул. Перенсона, 7 ауд. № 2-04	<p>Оборудование: Маркерная доска – 1 шт., ноутбук – 10шт., мультимедийный демонстрационный комплекс (проектор, интерактивная доска, колонки, USB-камера) – 1шт., система видеоконференцсвязи Policom – 1шт.</p> <p>Программное обеспечение Альт Образование 8 (лицензия № ААО.0006.00, договор № ДС 14-2017 от 27.12.2017</p>
Аудитории для практических (семинарских)/ лабораторных занятий	
Ул. Перенсона, 7 ауд. 2-04	<p>Оборудование: Маркерная доска – 1 шт., ноутбук – 10шт., мультимедийный демонстрационный комплекс (проектор, интерактивная доска, колонки, USB-камера) – 1шт., система видеоконференцсвязи Policom – 1шт.</p> <p>Программное обеспечение Альт Образование 8 (лицензия № ААО.0006.00, договор № ДС 14-2017 от 27.12.2017</p>
Ул. Перенсона, 7 ауд. 4-12	<p>Оборудование: Компьютер – 10 шт., проектор – 1 шт., интерактивная доска – 1шт., маркерная доска – 1 шт.</p> <p>Программное обеспечение Linux Mint – (Свободная лицензия GPL)</p>