ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Иркутский государственный университет путей сообщения»
(ФГБОУ ВО ИрГУПС)

УТВЕРЖДЕНА
приказом ректора
от «02» июня 2023 г. № 424-1

Б1.О.18 Программирование

рабочая программа дисциплины

Специальность/направление подготовки – 09.03.04 Программная инженерия
Специализация/профиль – Разработка программно-информационных систем
Квалификация выпускника – Бакалавр
Форма и срок обучения – очная форма 4 года
Кафедра-разработчик программы – Информационные системы и защита информации

Общая трудоемкость в з.е. – 18
Часов по учебному плану (УП) – 648

Формы промежуточной аттестации
очная форма обучения:
зачет 1, 2 семестр, экзамен 3 семестр, курсовая работа 3 семестр

Очная форма обучения **Распределение часов дисциплины по семестрам**

Семестр	1	2	3	**Итого**
Вид занятий	Часов по УП	Часов по УП	Часов по УП	**Часов по УП**
Аудиторная контактная работа по видам учебных занятий/ в т.ч. в форме ПП*	102	102	102	**306**
– лекции	34	34	34	**102**
– практические (семинарские)				
– лабораторные	68	68	68	**204**
Самостоятельная работа	114	114	78	**306**
Экзамен			36	**36**
Итого	216	216	216	**648**

ИРКУТСК

Электронный документ выгружен из ЕИС ФГБОУ ВО ИрГУПС и соответствует оригиналу
Подписант ФГБОУ ВО ИрГУПС Трофимов Ю.А.
00920FD815CE68F8C4CA795540563D259C с 07.02.2024 05:46 по 02.05.2025 05:46 GMT+03:00
Подпись соответствует файлу документа

Рабочая программа дисциплины разработана в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 09.03.04 Программная инженерия, утвержденным Приказом Министерства науки и высшего образования Российской Федерации от 19.09.2017 № 920.

Программу составил(и):
к.т.н., Доцент, Ю.Н. Шишкин
Ассистент, А.А. Шедиков

Рабочая программа рассмотрена и одобрена для использования в учебном процессе на заседании кафедры «Информационные системы и защита информации», протокол от «2» июня 2023 г. № 12

Зав. кафедрой, к. э. н, доцент Т.К. Кириллова

1 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ	
1.1 Цели дисциплины	
1	овладение теоретическими и прикладными профессиональными знаниями и умениями в области программирования на алгоритмических языках высокого уровня
2	приобретение навыков самостоятельного и творческого использования теоретических знаний в практической деятельности по производству программного обеспечения для информационно-вычислительных систем различного назначения
1.2 Задачи дисциплины	
1	овладение основными языками программирования высокого уровня
2	освоение основных технологий программирования на алгоритмических языках
3	освоение методики производства программного обеспечения для информационных систем
4	освоение методов производства программного обеспечения при коллективной его разработке
1.3 Цель воспитания и задачи воспитательной работы в рамках дисциплины	
Профессионально-трудовое воспитание обучающихся	
Цель – формирование у обучающихся осознанной профессиональной ориентации, понимания общественного смысла труда и значимости его для себя лично, ответственного, сознательного и творческого отношения к будущей деятельности, профессиональной этики, способности предвидеть изменения, которые могут возникнуть в профессиональной деятельности, и умению работать в изменённых, вновь созданных условиях труда. Цель профессионально-трудового воспитания достигается по мере решения в единстве следующих задач: – формирование сознательного отношения к выбранной профессии; – воспитание чести, гордости, любви к профессии, сознательного отношения к профессиональному долгу, понимаемому как личная ответственность и обязанность; – формирование психологии профессионала; – формирование профессиональной культуры, этики профессионального общения; – формирование социальной компетентности и другие задачи, связанные с имиджем профессии и авторитетом транспортной отрасли	

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	
Блок/часть ОПОП	Блок 1. Дисциплины / Обязательная часть
2.1 Дисциплины и практики, на которых основывается изучение данной дисциплины	
1	Б1.О.33 Схемотехнические основы программно-вычислительных систем
2	Б2.О.01(У) Учебная - ознакомительная практика
2.2 Дисциплины и практики, для которых изучение данной дисциплины необходимо как предшествующее	
1	Б1.О.13 Математическая логика и теория алгоритмов
2	Б1.О.21 Операционные системы
3	Б1.О.22 Базы данных
4	Б1.О.24 Компьютерные сети
5	Б1.О.25 Тестирование и отладка программного обеспечения
6	Б1.О.26 Объектно-ориентированное программирование
7	Б1.О.27 Надежность программного обеспечения
8	Б1.О.29 Вычислительные алгоритмы
9	Б1.О.30 Администрирование программно-информационных систем
10	Б1.О.31 Теория языков программирования и методы трансляции
11	Б1.О.32 Машинно-зависимые языки программирования
12	Б1.О.36 Проектирование программного обеспечения
13	Б2.О.01(У) Учебная - ознакомительная практика
14	Б2.О.02(У) Учебная - эксплуатационная практика
15	Б2.О.03(П) Производственная - технологическая (проектно-технологическая) практика
16	Б2.О.04(Пд) Производственная - преддипломная практика
17	Б3.01(Д) Выполнение выпускной квалификационной работы
18	Б3.02(Д) Защита выпускной квалификационной работы

3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ТРЕБОВАНИЯМИ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ		
Код и наименование компетенции	**Код и наименование индикатора достижения компетенции**	**Планируемые результаты обучения**
ОПК-2 Способен понимать принципы работы современных	ОПК-2.1 Знает современные информационные технологии и программные	Знать: основные парадигмы программирования, номенклатуру языков программирования различного уровня и их возможности

информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	средства, в том числе отечественного производства при решении задач профессиональной деятельности	Уметь: разрабатывать и записывать алгоритмы и программы на языке высокого уровня в соответствие с принципом структурности
		Владеть: основными методами, способами и средствами переработки информации на основе парадигмы императивного программирования
	ОПК-2.2 Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знать: основные понятия языков высокого уровня программирования и их классификацию: идентификаторы, данные и их типы операции, стандартные функции, выражения, операторы
		Уметь: описывать алфавит и основные синтаксические конструкции языка Си
		Владеть: навыками проектирования, ввода, отладки и тестирования программ
	ОПК-2.3 Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Знать: основные синтаксические конструкции, используемые в языке программирования для описания и обработки данных типа массив
		Уметь: разрабатывать алгоритмы и программы обработки массивов
		Владеть: навыками решения учебных задач в соответствие с принципом модульности средствами языка Си, PYTHON, Golang
ОПК-5 Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем	Знать: основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем
		Уметь: ставить и решать задачи администрирования информационных систем
		Владеть: навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
	ОПК-5.2 Умеет выполнять параметрическую настройку информационных и автоматизированных систем	Знать: параметрическую настройку информационных и автоматизированных систем
		Уметь: анализировать техническую документацию, производить настройку, наладку и тестирование информационных и автоматизированных систем
		Владеть: методами настройки, наладки программно-аппаратных комплексов
	ОПК-5.3 Имеет навыки инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем	Знать: общие принципы организации операционного программного обеспечения сетевого оборудования
		Уметь: сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем
		Владеть: навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1 Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий	Знать: базы данных, языки программирования и среды программирования
		Уметь: эффективно применять в решении профессиональных задач языки программирования, базы данных и среды программирования
		Владеть: приемами программирования, отладки и тестирования прототипов программно-технических комплексов задач
	ОПК-6.2 Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения	Знать: языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий
		Уметь: применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ

	прикладных задач различных классов, ведения баз данных и информационны	Владеть: навыками использования современного программного обеспечения для решения прикладных задач различных классов
	ОПК-6.3 Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	Знать: языки программирования, базы данных и среды программирования
		Уметь: применять методы программирования, отладки и тестирования прототипов программно-технических комплексов задач
		Владеть: навыками программирования на языках высокого уровня, реализующие принципы объектно-ориентированного программирования; методами функционального тестирования прототипов программно-технических комплексов

4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Код	Наименование разделов, тем и видов работ	Очная форма					*Код индикатора достижения компетенции
		Семестр	Часы				
			Лек	Пр	Лаб	СР	
1.0	**Раздел 1. Программирование на языке высокого уровня C++.**						
1.1	Тема 1. Алфавит языка. Служебные слова. Константы. Переменные. Стандартные функции	1	4		8	8	ОПК-2.1
1.2	Тема 2. Выражения. Операторы присваивания. Структура программы	1	4		8	8	ОПК-2.2
1.3	Тема 3. Ввод и вывод данных	1	2		8	4	ОПК-2.3
1.4	Тема 4. Алгоритм и его свойства. Схема алгоритма. Базовые структуры языка. Цепочка. Ветвления. Циклы.	1	8		16	35	ОПК-5.1
1.5	Тема 5. Структуры данных. Массивы. Строки. Множества. Записи. Файлы.	1	8		14	40	ОПК-5.2
1.6	Тема 6. Подпрограммы. Функции. Процедуры. Рекурсия.	1	4		8	10	ОПК-5.3
1.7	Тема 7. Ссылки и динамические переменные.	1	4		6	4	ОПК-6.1 ОПК-6.2 ОПК-6.3
	Форма промежуточной аттестации – зачет	1					ОПК-2.1 ОПК-2.2 ОПК-2.3
2.0	**Раздел 2. Современные языки программирования (Golang).**						
2.1	Тема 8. Основы языков высокого уровня (Golang). Алфавит языков. Константы. Переменные. Математические функции.	2	4		6	8	ОПК-2.1
2.2	Тема 9. Выражения и операторы присваивания.	2	4		8	8	ОПК-2.2
2.3	Тема 10. Ввод и вывод данных.	2	2		6	8	ОПК-2.3
2.4	Тема 11. Структура программы. Директивы препроцессора.	2	2		10	4	ОПК-5.1
2.5	Тема 12. Базовые структуры языков: цепочка, ветвления, циклы.	2	6		14	30	ОПК-5.2
2.6	Тема 13. Структуры данных: массивы, строки, файлы.	2	8		14	40	ОПК-5.3
2.7	Тема 14. Функции	2	4		10	10	ОПК-6.1
2.8	Тема 15. Адреса и указатели. Указатели и массивы. Указатели и строки.	2	4		8	6	ОПК-6.2 ОПК-6.3
	Форма промежуточной аттестации – зачет	2					ОПК-5.1 ОПК-5.2
3.0	**Раздел 3. Современные языки программирования (PYTHON).**						
3.1	Тема 16. Особенности программирования на языке Python. Среда программирования.	3	2		4	4	ОПК-2.1
3.2	Тема 17. Алфавит языков. Типы и модели данных. Константы. Переменные. Математические функции.	3	6		8	4	ОПК-2.2
3.3	Тема 18. Выражения и операторы присваивания.	3	2		8	4	ОПК-2.3

4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Код	Наименование разделов, тем и видов работ	Очная форма					*Код индикатора достижения компетенции
		Семестр	Часы				
			Лек	Пр	Лаб	СР	
3.4	Тема 19. Базовые структуры языка: цепочка, ветвления, циклы.	3	6		10	16	ОПК-5.1
3.5	Тема 20. Структуры данных: списки, кортежи, массивы, строки, множества, словари, файлы.	3	8		12	5	ОПК-5.2
3.6	Тема 21. Функции. Модули.	3	2		6	4	ОПК-5.3
3.7	Тема 22. Обработка исключений.	3	2		6	2	ОПК-6.1
3.8	Тема 23. Классы и объекты.	3	6		6	4	ОПК-6.2 ОПК-6.3
	Форма промежуточной аттестации – экзамен	3	36				ОПК-6.1 ОПК-6.2 ОПК-6.3
	Курсовая работа	3				40	ОПК-6.1 ОПК-6.2
	Итого часов (без учёта часов на промежуточную аттестацию)		102		204	306	

5 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине оформлен в виде приложения № 1 к рабочей программе дисциплины и размещен в электронной информационно-образовательной среде Университета, доступной обучающемуся через его личный кабинет

6 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1 Учебная литература

6.1.1 Основная литература

	Библиографическое описание	Кол-во экз. в библиотеке/ онлайн
6.1.1.1	Объектно-ориентированное программирование: учебное пособие (практикум) : направление подготовки 20.04.01 техносферная безопасность; направленность (профиль) «защита в чрезвычайных ситуациях». — Ставрополь : СКФУ, 2018. — 119 с. — URL: https://e.lanbook.com/book/306956 (дата обращения: 15.04.2024). — Текст : электронный.	Онлайн
6.1.1.2	Моделирование и визуализация экспериментальных данных: лабораторный практикум : учебное пособие: направление подготовки 09.03.02 информационные системы и технологии; направленность (профили) "прикладное программирование в информационных системах". бакалавриат. — Ставрополь : СКФУ, 2018. — 126 с. — URL: https://e.lanbook.com/book/306950 (дата обращения: 15.04.2024). — Текст : электронный.	Онлайн
6.1.1.3	Методические указания для выполнения курсовой работы по дисциплине «Программирование» для обучающихся очной и заочной форм обучения по направлению подготовки 38.03.02 – Менеджмент и специальности 38.05.01 – Экономическая безопасность для обучающихся направления 09.03.01 Информатика и вычислительная техника, профиль – Системы автоматизированного проектирования : методические указания. — Воронеж : Мичуринский ГАУ, 2019. — 12 с. — URL: https://e.lanbook.com/book/157822 (дата обращения: 15.04.2024). — Текст : электронный.	Онлайн

6.1.2 Дополнительная литература

	Библиографическое описание	Кол-во экз. в библиотеке/ онлайн
6.1.2.1	Основы программирования : методическое пособие для студентов 2-го курса специальности 09.02.07 «информационные системы и программирование». — Сочи : СГУ, 2019. — 52 с. — URL: https://e.lanbook.com/book/147661 (дата обращения: 15.04.2024). — Текст : электронный.	Онлайн

6.1.2.2	Сборка и программирование робота «Robotis Mini» : учебно-методический комплекс. — Липецк : Липецкий ГПУ, 2019. — 95 с. — URL: https://e.lanbook.com/book/146729 (дата обращения: 15.04.2024). — Текст : электронный.	Онлайн
6.1.2.3	Авдеенко, Т. В. Программирование в среде Visual Prolog : учебное пособие / Т. В. Авдеенко, М. Ю. Целебровская. — Новосибирск : НГТУ, 2022. — 46 с. — URL: https://e.lanbook.com/book/306320 (дата обращения: 15.04.2024). — Текст : электронный.	Онлайн
6.1.3 Учебно-методические разработки (в т. ч. для самостоятельной работы обучающихся)		
	Библиографическое описание	Кол-во экз. в библиотеке/ онлайн
6.1.3.1	Шишкин Ю.Н., Шедиков А.А. Методические указания по изучению дисциплины Б1.О.18 Программирование по направлению подготовки 09.03.04 Программная инженерия, профиль Разработка программно-информационных систем / Ю.Н. Шишкин; ИрГУПС. – Иркутск : ИрГУПС, 2023. – 13 с - Текст: электронный. - URL: https://www.irgups.ru/eis/for_site/umkd_files/mu_9337_1398_2023_1_signed.pdf	Онлайн
6.2 Ресурсы информационно-телекоммуникационной сети «Интернет»		
6.2.1	Электронно-библиотечная система «Издательство Лань», https://e.lanbook.com/	
6.3 Программное обеспечение и информационные справочные системы		
6.3.1 Базовое программное обеспечение		
6.3.1.1	Microsoft Windows Professional 10, государственный контракт от 20.07.2021 № 0334100010021000013-01	
6.3.1.2	Microsoft Office Russian 2010, государственный контракт от 20.07.2021 № 0334100010021000013-01	
6.3.1.3	FoxitReader, свободно распространяемое программное обеспечение http://free-software.com.ua/pdf-viewer/foxit-reader/	
6.3.1.4	Adobe Acrobat Reader DC свободно распространяемое программное обеспечение https://get.adobe.com/ru/reader/enterprise/	
6.3.1.5	Яндекс. Браузер. Прикладное программное обеспечение общего назначения, Офисные приложения, лицензия – свободно распространяемое программное обеспечение по лицензии BSD License	
6.3.2 Специализированное программное обеспечение		
6.3.2.1	Python 3.9, свободно распространяемое программное обеспечение https://docs.python.org/3/license.html	
6.3.2.2	Dev-C++, свободная интегрированная среда разработки приложений для языков программирования C/C++, https://code-live.ru/post/dev-cpp-free-cpp-ide-for-windows/	
6.3.2.3	Visual Studio 2022 Community, образовательная лицензия, https://visualstudio.microsoft.com/ru/license-terms/vs2022-ga-community/	
6.3.2.4	Visual Studio Code, образовательная лицензия, https://code.visualstudio.com/license	
6.3.2.5	NetBeans IDE, свободная лицензия Apache License 2.0 https://www.apache.org/licenses/	
6.3.2.6	Java Virtual Machine, свободная лицензия Oracle Java SE https://www.oracle.com/downloads/licenses/javase-license1.html	
6.3.3 Информационные справочные системы		
6.3.3.1	Не предусмотрены	
6.4 Правовые и нормативные документы		
6.4.1	Не предусмотрены	

7 ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ УЧЕБНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

1	Корпуса А, Б, В, Г, Д, Е ИрГУПС находятся по адресу г. Иркутск, ул. Чернышевского, д. 15; корпус Л ИрГУПС находится – по адресу г. Иркутск, ул. Лермонтова, д.80
2	Учебная аудитория Д-313 для проведения лекционных и практических занятий, групповых и индивидуальных консультаций, курсового проектирования (выполнения курсовых работ), текущего контроля и промежуточной аттестации. Основное оборудование: специализированная мебель, мультимедиапроектор, экран, ноутбук (переносной). Для проведения занятий имеются учебно-наглядные пособия (презентации).
3	Компьютерный класс «Информатика». «Информационные технологии» Д-505 для проведения практических занятий, лабораторных работ, групповых и индивидуальных консультаций, курсового проектирования (выполнения курсовых работ), текущего контроля и промежуточной аттестации. Основное оборудование: специализированная мебель, компьютеры с подключением к сети Интернет, обеспечивающие доступ в электронную информационно-образовательную среду ИрГУПС,

	мультимедиапроектор (переносной), экран (переносной), компьютер. Для проведения занятий имеются учебно-наглядные пособия (презентации, плакаты).
4	Компьютерный класс «Информатика». «Технологии и методы программирования» Д-503 для проведения практических занятий, лабораторных работ, групповых и индивидуальных консультаций, курсового проектирования (выполнения курсовых работ), текущего контроля и промежуточной аттестации. Основное оборудование: специализированная мебель, компьютеры с подключением к сети Интернет, обеспечивающие доступ в электронную информационно-образовательную среду ИрГУПС, мультимедиапроектор (переносной), экран (переносной), компьютер. Для проведения занятий имеются учебно-наглядные пособия (презентации, плакаты).
5	Компьютерный класс (тестирование студентов) Д-507 для проведения практических занятий, лабораторных работ, групповых и индивидуальных консультаций, курсового проектирования (выполнения курсовых работ), текущего контроля и промежуточной аттестации. Основное оборудование: специализированная мебель, компьютеры с подключением к сети Интернет, обеспечивающие доступ в электронную информационно-образовательную среду ИрГУПС, мультимедиапроектор (переносной), экран (переносной), компьютер. Для проведения занятий имеются учебно-наглядные пособия (презентации, плакаты).
6	Компьютерный класс «Информатика». «Технологии и методы программирования» Д-503 для проведения практических занятий, лабораторных работ, групповых и индивидуальных консультаций, курсового проектирования (выполнения курсовых работ), текущего контроля и промежуточной аттестации. Основное оборудование: Для проведения занятий имеются учебно-наглядные пособия (презентации, плакаты).
7	Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, подключенной к информационно-телекоммуникационной сети «Интернет», и обеспечены доступом в электронную информационно-образовательную среду ИрГУПС. Помещения для самостоятельной работы обучающихся: – читальные залы; – учебные залы вычислительной техники А-401, А-509, А-513, А-516, Д-501, Д-503, Д-505, Д-507; – помещения для хранения и профилактического обслуживания учебного оборудования – А-521

8 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Вид учебной деятельности	Организация учебной деятельности обучающегося
Лекция	Лекция (от латинского «lection» – чтение) – вид аудиторных учебных занятий. Лекция: закладывает основы научных знаний в систематизированной, последовательной, обобщенной форме; раскрывает состояние и перспективы развития соответствующей области науки и техники; концентрирует внимание обучающихся на наиболее сложных, узловых вопросах; стимулирует познавательную активность обучающихся. Во время лекционных занятий обучающийся должен уметь сконцентрировать внимание на изучаемых проблемах и включить в работу все виды памяти: словесную, образную и моторно-двигательную. Для этого весь материал, излагаемый преподавателем, обучающемуся необходимо конспектировать. На полях конспекта следует помечать вопросы, выделенные обучающимся для консультации с преподавателем. Выводы, полученные в виде формул, рекомендуется в конспекте подчеркивать или обводить рамкой, чтобы лучше запоминались. Полезно составить краткий справочник, содержащий определения важнейших понятий лекции. К каждому занятию следует разобрать материал предыдущей лекции. Изучая материал по учебнику или конспекту лекций, следует переходить к следующему вопросу только в том случае, когда хорошо усвоен предыдущий вопрос. Ряд вопросов дисциплины может быть вынесен на самостоятельное изучение. Такое задание требует оперативного выполнения. В конспекте лекций необходимо оставить место для освещения упомянутых вопросов. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, то необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии
Практическое занятие	Практическое занятие – вид аудиторных учебных занятий, целенаправленная форма организации учебного процесса, при реализации которой обучающиеся по заданию и под руководством преподавателя выполняют практические задания. Практические задания направлены на углубление научно-теоретических знаний и овладение определенными методами работы, в процессе которых вырабатываются умения и навыки выполнения тех или иных учебных действий в данной сфере науки. Практические занятия развивают научное мышление и речь, позволяют проверить знания обучающихся, выступают как средства оперативной обратной связи; цель практических занятий – углублять, расширять, детализировать знания, полученные на лекции, в обобщенной форме и содействовать выработке навыков профессиональной деятельности.

	На практических занятиях подробно рассматриваются основные вопросы дисциплины, разбираются основные типы задач. К каждому практическому занятию следует заранее самостоятельно выполнить домашнее задание и выучить лекционный материал к следующей теме. Систематическое выполнение домашних заданий обязательно и является важным фактором, способствующим успешному усвоению дисциплины
Лабораторная работа	Основной целью лабораторных работ является теоретическое обоснование, наглядное и/или экспериментальное подтверждение и/или проверка существенных теоретических положений (законов, закономерностей) анализ существующих методик и методов их реализации и т.д. Они занимают преимущественное место при изучении дисциплин обязательной части и части, формируемой участниками образовательных отношений Блока 1. Исходя из цели, содержанием лабораторных работ могут быть: - экспериментальная проверка формул, методик расчета; - проведение натурных измерений свойств, рабочих параметров, режимов работы при помощи лабораторного оборудования и/или стендов и макетов; - ознакомление, анализ и теоретические выкладки по устройству, принципу действия и способам обслуживания аппаратов, деталей машин, механизмов, процессов, протекающих в них при этом и т.д.; - наглядная графическая интерпретация чертежей, схем, объемных поверхностей и т.д., воспроизводимых с помощью специализированного программного обеспечения; - имитационное моделирование процессов, протекающих в сложных химических, физических, механических, электрических и пр. объектах; - наглядное представление о работе персонала конкретной организации или подразделения ОАО «РЖД» посредством моделирования штатных и внештатных ситуаций в виртуальных специализированных АРМ (автоматизированных рабочих мест); - установление и подтверждение закономерностей (путем сравнения проведенного эксперимента и рассчитанных значений) и т.д.; - ознакомление с методиками проведения экспериментов, наглядным устройством стенд-макетов и пр.; - установление свойств веществ, их качественных и количественных характеристик; - анализ различных характеристик процессов, в том числе производственных и иных процессов; - расчет параметров различных явлений и процессов, смоделировать которые не возможно в реальных условиях (например, чрезвычайные ситуации и пр.); - наблюдение развития явлений, процессов и др. Допускается иное содержание лабораторных работ, если это будет способствовать реализации целей и задач дисциплины и формированию соответствующих компетенций. По характеру выполняемых лабораторных работ возможны: - ознакомительные работы, используемые для закрепления изученного теоретического материалы; - аналитические работы, используемые для получения новой информации на основе формализованных методов; - творческие работы, ориентированные на самостоятельный выбор подходов решения задач. Прежде, чем приступить к лабораторным занятиям, обучающимся необходимо повторить теоретический материал по теме работы. Каждая лабораторная работа оснащена методическими указаниями, разработанными преподавателями, ведущими дисциплину
Самостоятельная работа	Обучение по дисциплине «Программирование» предусматривает активную самостоятельную работу обучающегося. В разделе 4 рабочей программы, который называется «Структура и содержание дисциплины», все часы самостоятельной работы расписаны по темам и вопросам, а также указана необходимая учебная литература: обучающийся изучает учебный материал, разбирает примеры и решает разноуровневые задачи в рамках выполнения как общих домашних заданий, так и индивидуальных домашних заданий (ИДЗ) и других видов работ, предусмотренных рабочей программой дисциплины. При выполнении домашних заданий обучающемуся следует обратиться к задачам, решенным на предыдущих практических занятиях, решенным домашним работам, а также к примерам, приводимым лектором. Если этого будет недостаточно для выполнения всей работы можно дополнительно воспользоваться учебными пособиями, приведенными в разделе 6.1 «Учебная литература». Если, несмотря на изученный материал, задание выполнить не удается, то в обязательном порядке необходимо посетить консультацию преподавателя, ведущего практические занятия, и/или консультацию лектора. Домашние задания, индивидуальные домашние задания и другие работы, предусмотренные рабочей программой дисциплины должны быть выполнены обучающимся в установленные преподавателем сроки в соответствии с требованиями к оформлению

	текстовой и графической документации, сформулированным в Положении «Требования к оформлению текстовой и графической документации. Нормоконтроль»
Комплекс учебно-методических материалов по всем видам учебной деятельности, предусмотренным рабочей программой дисциплины (модуля), размещен в электронной информационно-образовательной среде ИрГУПС, доступной обучающемуся через его личный кабинет	

Приложение № 1 к рабочей программе

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля успеваемости и промежуточной аттестации

1. Общие положения

Фонд оценочных средств (ФОС) является составной частью нормативно-методического обеспечения системы оценки качества освоения обучающимися образовательной программы.

Фонд оценочных средств предназначен для использования обучающимися, преподавателями, администрацией ИрГУПС, а также сторонними образовательными организациями для оценивания качества освоения образовательной программы и уровня сформированности компетенций у обучающихся.

Задачами ФОС являются:

– оценка достижений обучающихся в процессе изучения дисциплины;

– обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности через совершенствование традиционных и внедрение инновационных методов обучения в образовательный процесс;

– самоподготовка и самоконтроль обучающихся в процессе обучения.

Фонд оценочных средств сформирован на основе ключевых принципов оценивания: валидность, надежность, объективность, эффективность.

Для оценки уровня сформированности компетенций используется трехуровневая система:

– минимальный уровень освоения, обязательный для всех обучающихся по завершению освоения образовательной программы; дает общее представление о виде деятельности, основных закономерностях функционирования объектов профессиональной деятельности, методов и алгоритмов решения практических задач;

– базовый уровень освоения, превышение минимальных характеристик сформированности компетенций; позволяет решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам;

– высокий уровень освоения, максимально возможная выраженность характеристик компетенций; предполагает готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении.

2. Перечень компетенций, в формировании которых участвует дисциплина.
Программа контрольно-оценочных мероприятий.
Показатели оценивания компетенций, критерии оценки

Дисциплина «Программирование» участвует в формировании компетенций:

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности

ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

Программа контрольно-оценочных мероприятий очная форма обучения

№	Наименование контрольно-оценочного мероприятия	Объект контроля	Код индикатора достижения компетенции	Наименование оценочного средства (форма проведения*)
		1 семестр		
1.0	**Раздел 1. Программирование на языке высокого уровня С++**			
1.1	Текущий контроль	Тема 1. Алфавит языка. Служебные слова. Константы. Переменные. Стандартные функции	ОПК-2.1	Лабораторная работа (письменно/устно)
1.2	Текущий контроль	Тема 2. Выражения. Операторы присваивания. Структура программы	ОПК-2.2	Лабораторная работа (письменно/устно)
1.3	Текущий контроль	Тема 3. Ввод и вывод данных	ОПК-2.3	Лабораторная работа (письменно/устно)
1.4	Текущий контроль	Тема 4. Алгоритм и его свойства. Схема алгоритма. Базовые структуры языка. Цепочка. Ветвления. Циклы.	ОПК-5.1	Лабораторная работа (письменно/устно)
1.5	Текущий контроль	Тема 5. Структуры данных. Массивы. Строки. Множества. Записи. Файлы.	ОПК-5.2	Лабораторная работа (письменно/устно)
1.6	Текущий контроль	Тема 6. Подпрограммы. Функции. Процедуры. Рекурсия.	ОПК-5.3	Лабораторная работа (письменно/устно)
1.7	Текущий контроль	Тема 7. Ссылки и динамические переменные.	ОПК-6.1 ОПК-6.2 ОПК-6.3	Лабораторная работа (письменно/устно)
	Промежуточная аттестация		ОПК-2.1 ОПК-2.2 ОПК-2.3	Зачет (собеседование) Зачет - тестирование (компьютерные технологии)
		2 семестр		
2.0	**Раздел 2. Современные языки программирования (Golang)**			
2.1	Текущий контроль	Тема 8. Основы языков высокого уровня (Golang). Алфавит языков. Константы. Переменные. Математические функции.	ОПК-2.1	Лабораторная работа (письменно/устно)
2.2	Текущий контроль	Тема 9. Выражения и операторы присваивания.	ОПК-2.2	Лабораторная работа (письменно/устно)
2.3	Текущий контроль	Тема 10. Ввод и вывод данных.	ОПК-2.3	Лабораторная работа (письменно/устно)
2.4	Текущий контроль	Тема 11. Структура программы. Директивы препроцессора.	ОПК-5.1	Лабораторная работа (письменно/устно)
2.5	Текущий контроль	Тема 12. Базовые структуры языков: цепочка, ветвления, циклы.	ОПК-5.2	Лабораторная работа (письменно/устно)

2.6	Текущий контроль	Тема 13. Структуры данных: массивы, строки, файлы.	ОПК-5.3	Лабораторная работа (письменно/устно)
2.7	Текущий контроль	Тема 14. Функции	ОПК-6.1	Лабораторная работа (письменно/устно)
2.8	Текущий контроль	Тема 15. Адреса и указатели. Указатели и массивы. Указатели и строки.	ОПК-6.2 ОПК-6.3	Лабораторная работа (письменно/устно)
	Промежуточная аттестация		ОПК-5.1 ОПК-5.2	Зачет (собеседование) Зачет - тестирование (компьютерные технологии)
3 семестр				
3.0	**Раздел 3. Современные языки программирования (PYTHON)**			
3.1	Текущий контроль	Тема 16. Особенности программирования на языке Python. Среда программирования.	ОПК-2.1	Лабораторная работа (письменно/устно)
3.2	Текущий контроль	Тема 17. Алфавит языков. Типы и модели данных. Константы. Переменные. Математические функции.	ОПК-2.2	Лабораторная работа (письменно/устно)
3.3	Текущий контроль	Тема 18. Выражения и операторы присваивания.	ОПК-2.3	Лабораторная работа (письменно/устно)
3.4	Текущий контроль	Тема 19. Базовые структуры языка: цепочка, ветвления, циклы.	ОПК-5.1	Лабораторная работа (письменно/устно)
3.5	Текущий контроль	Тема 20. Структуры данных: списки, кортежи, массивы, строки, множества, словари, файлы.	ОПК-5.2	Лабораторная работа (письменно/устно)
3.6	Текущий контроль	Тема 21. Функции. Модули.	ОПК-5.3	Лабораторная работа (письменно/устно)
3.7	Текущий контроль	Тема 22. Обработка исключений.	ОПК-6.1	Лабораторная работа (письменно/устно)
3.8	Текущий контроль	Тема 23. Классы и объекты.	ОПК-6.2 ОПК-6.3	Лабораторная работа (письменно/устно)
	Промежуточная аттестация		ОПК-6.1 ОПК-6.2	Курсовая работа (письменно) Курсовая работа (устно)
	Промежуточная аттестация		ОПК-6.1 ОПК-6.2 ОПК-6.3	Экзамен (собеседование) Экзамен - тестирование (компьютерные технологии)

*Форма проведения контрольно-оценочного мероприятия: устно, письменно, компьютерные технологии.

Описание показателей и критериев оценивания компетенций. Описание шкал оценивания

Контроль качества освоения дисциплины включает в себя текущий контроль успеваемости и промежуточную аттестацию. Текущий контроль успеваемости и промежуточная аттестация обучающихся проводятся в целях установления соответствия достижений обучающихся поэтапным требованиям образовательной программы к результатам обучения и формирования компетенций.

Текущий контроль успеваемости – основной вид систематической проверки знаний, умений, навыков обучающихся. Задача текущего контроля – оперативное и регулярное управление учебной деятельностью обучающихся на основе обратной связи и корректировки. Результаты оценивания учитываются в виде средней оценки при проведении промежуточной аттестации.

Для оценивания результатов обучения используется четырехбалльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» и двухбалльная шкала: «зачтено», «не зачтено».

Перечень оценочных средств, используемых для оценивания компетенций, а также краткая характеристика этих средств приведены в таблице.

Текущий контроль

№	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Лабораторная работа	Средство, позволяющее оценить умение обучающегося письменно/устно излагать суть поставленной задачи, самостоятельно применять стандартные методы решения поставленной задачи с использованием имеющейся лабораторной базы, проводить анализ полученного результата работы. Может быть использовано для оценки умений, навыков и (или) опыта деятельности обучающихся	Образец задания для выполнения лабораторной работы и примерный перечень вопросов для ее защиты

Промежуточная аттестация

№	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Зачет	Средство, позволяющее оценить знания, умения, навыков и (или) опыта деятельности обучающегося по дисциплине. Может быть использовано для оценки знаний, умений, навыков и (или) опыта деятельности обучающихся	Перечень теоретических вопросов и практических заданий к зачету
2	Тест – промежуточная аттестация в форме зачета	Система автоматизированного контроля освоения компетенций (части компетенций) обучающимся по дисциплине (модулю) с использованием информационно-коммуникационных технологий. Может быть использовано для оценки знаний, умений, навыков и (или) опыта деятельности обучающихся	Фонд тестовых заданий
3	Экзамен	Средство, позволяющее оценить знания, умения, навыков и (или) опыта деятельности обучающегося по дисциплине. Может быть использовано для оценки знаний, умений, навыков и (или) опыта деятельности обучающихся	Перечень теоретических вопросов и практических заданий (образец экзаменационного билета) к экзамену
4	Тест – промежуточная аттестация в форме экзамена	Система автоматизированного контроля освоения компетенций (части компетенций) обучающимся по дисциплине (модулю) с использованием информационно-коммуникационных технологий. Может быть использовано для оценки знаний, умений, навыков и (или) опыта деятельности обучающихся	Фонд тестовых заданий
5	Курсовая работа	Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся. Может быть использовано для оценки умений, навыков и (или) опыта деятельности обучающихся в предметной или межпредметной областях	Образец задания для выполнения курсовой работы и примерный перечень вопросов для ее защиты

Критерии и шкалы оценивания компетенций в результате изучения дисциплины при проведении промежуточной аттестации в форме зачета и экзамена. Шкала оценивания уровня освоения компетенций

Шкалы оценивания		Критерии оценивания	Уровень освоения компетенции
«отлично»	«зачтено»	Обучающийся правильно ответил на теоретические вопросы. Показал отличные знания в рамках учебного материала. Правильно выполнил практические задания. Показал отличные умения и владения навыками применения полученных знаний и умений при решении задач в рамках учебного материала. Ответил на все дополнительные вопросы	Высокий
«хорошо»		Обучающийся с небольшими неточностями ответил на теоретические вопросы. Показал хорошие знания в рамках учебного материала. С небольшими неточностями выполнил практические задания. Показал хорошие умения и владения навыками применения полученных знаний и умений при решении задач в рамках учебного материала. Ответил на большинство дополнительных вопросов	Базовый
«удовлетворительно»		Обучающийся с существенными неточностями ответил на теоретические вопросы. Показал удовлетворительные знания в рамках учебного материала. С существенными неточностями выполнил практические задания. Показал удовлетворительные умения и владения навыками применения полученных знаний и умений при решении задач в рамках учебного материала. Допустил много неточностей при ответе на дополнительные вопросы	Минимальный
«неудовлетворительно»	«не зачтено»	Обучающийся при ответе на теоретические вопросы и при выполнении практических заданий продемонстрировал недостаточный уровень знаний и умений при решении задач в рамках учебного материала. При ответах на дополнительные вопросы было допущено множество неправильных ответов	Компетенция не сформирована

Тест – промежуточная аттестация в форме зачета и экзамена

Шкала оценивания		Критерии оценивания
«отлично»	«зачтено»	Обучающийся верно ответил на 90 – 100 % тестовых заданий при прохождении тестирования
«хорошо»		Обучающийся верно ответил на 80 – 89 % тестовых заданий при прохождении тестирования
«удовлетворительно»		Обучающийся верно ответил на 70 – 79 % тестовых заданий при прохождении тестирования
«неудовлетворительно»	«не зачтено»	Обучающийся верно ответил на 69 % и менее тестовых заданий при прохождении тестирования

Курсовая работа

Шкала оценивания	Критерии оценивания
«отлично»	Содержание курсовой работы полностью соответствует заданию. Представлены результаты обзора литературных и иных источников. Структура курсовой работы логически и методически выдержана. Все выводы и предложения убедительно аргументированы. Оформление курсовой работы и полученные результаты полностью отвечают требованиям, изложенным в методических указаниях. При защите курсовой работы обучающийся правильно и уверенно отвечает на вопросы

	преподавателя, демонстрирует глубокое знание теоретического материала, способен аргументировать собственные утверждения и выводы
«хорошо»	Содержание курсовой работы полностью соответствует заданию. Представлены результаты обзора литературных и иных источников. Структура курсовой работы логически и методически выдержана. Большинство выводов и предложений аргументировано. Оформление курсовой работы и полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две несущественные ошибки в использовании терминов, в построенных диаграммах и схемах. Наличествует незначительное количество грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся правильно и уверенно отвечает на большинство вопросов преподавателя, демонстрирует хорошее знание теоретического материала, но не всегда способен аргументировать собственные утверждения и выводы. При наводящих вопросах преподавателя исправляет ошибки в ответе
«удовлетворительно»	Содержание курсовой работы частично не соответствует заданию. Результаты обзора литературных и иных источников представлены недостаточно полно. Есть нарушения в логике изложения материала. Аргументация выводов и предложений слабая или отсутствует. Имеются одно-два существенных отклонений от требований в оформлении курсовой работы. Полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две существенных ошибки в использовании терминов, в построенных диаграммах и схемах. Много грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся допускает грубые ошибки при ответах на вопросы преподавателя и /или не дал ответ более чем на 30% вопросов, демонстрирует слабое знание теоретического материала, в большинстве случаев не способен уверенно аргументировать собственные утверждения и выводы
«неудовлетворительно»	Содержание курсовой работы в целом не соответствует заданию. Имеются более двух существенных отклонений от требований в оформлении курсовой работы. Большое количество существенных ошибок по сути работы, много грамматических и стилистических ошибок и др. Полученные результаты не отвечают требованиям, изложенным в методических указаниях. При защите курсовой работы обучающийся демонстрирует слабое понимание программного материала. Курсовая работа не представлена преподавателю. Обучающийся не явился на защиту курсовой работы

Критерии и шкалы оценивания результатов обучения при проведении текущего контроля успеваемости

Лабораторная работа

Шкалы оценивания		Критерии оценивания
«отлично»	«зачтено»	Лабораторная работа выполнена в обозначенный преподавателем срок, письменный отчет без замечаний. Лабораторная работа выполнена обучающимся в полном объеме с соблюдением необходимой последовательности. Обучающийся работал полностью самостоятельно; показал необходимые для проведения работы теоретические знания, практические умения и навыки. Работа (отчет) оформлена аккуратно, в наиболее оптимальной для фиксации результатов форме
«хорошо»		Лабораторная работа выполнена в обозначенный преподавателем срок, письменный отчет с небольшими недочетами. Лабораторная работа выполнена обучающимся в полном объеме и самостоятельно. Допущены отклонения от необходимой последовательности выполнения, не влияющие на правильность конечного результата. Работа показывает знание обучающимся основного теоретического материала и овладение умениями, необходимыми для самостоятельного выполнения работы. Допущены неточности и небрежность в оформлении результатов работы (отчета)
«удовлетворительно»		Лабораторная работа выполнена с задержкой, письменный отчет с недочетами. Лабораторная работа выполняется и оформляется обучающимся при посторонней помощи. На выполнение работы затрачивается много времени. Обучающийся показывает знания теоретического

		материала, но испытывает затруднение при самостоятельной работе с источниками знаний или приборами
«неудовлетворительно»	«не зачтено»	Лабораторная работа не выполнена, письменный отчет не представлен. Результаты, полученные обучающимся, не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Показывается плохое знание теоретического материала и отсутствие необходимых умений. Лабораторная работа не выполнена, у учащегося отсутствуют необходимые для проведения работы теоретические знания, практические умения и навыки

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности

3.1 Типовые задания для выполнения лабораторной работы и примерный перечень вопросов для ее защиты

Контрольные варианты заданий выложены в электронной информационно-образовательной среде ИрГУПС, доступной обучающемуся через его личный кабинет.

Ниже приведен образец типовых вариантов заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты.

«Решение задач линейной и разветвляющейся структуры»

Задание: Составить программу по варианту

С клавиатуры вводится четырёхзначное число. Найти
- сумму первой и последней цифры числа;
- произведение второй и третьей цифры числа.

Вычислить значение функции

$$t = \frac{2\cos(x - \pi/6)}{0{,}5 + \sin^2 y}\left(1 + \frac{z^2}{3 - z^2/5}\right).$$

При $x = 14.26$, $y = -1.22$, $z = 3.5\times10^{-2}$: **0.564846.**

Шахматная доска содержит 8 × 8 клеток. Горизонтали и вертикали обозначаются цифрами от 1 – 8. Написать программу, проверяющую, можно ли с первой заданной клетки попасть на вторую одним ходом ладьи.
Вычислить значение Y в зависимости от выбранной функции $\varphi(x)$, аргумент которой определяется из поставленного условия. Возможные значения функции $\varphi(x)$: 2x, x^3, x/2. Пользователь должен указать при какой функции $\varphi(x)$ программа будет выполнять вычисления. Предусмотреть вывод сообщений, показывающих, при каком условии и с какой функцией производились вычисления Y. Предусмотреть диалог с пользователем и дружественный интерфейс. $y = 2a\cos^3 x^2 + \sin^2 x^3 - b\varphi(x)$, где $x = \begin{cases} z^3 + 0{,}2; & z < 1; \\ z + \ln z; & z \geq 1. \end{cases}$ Составить блок-схему к программе.

Контрольные вопросы:

1) Опишите основные характеристики языка Python
2) Что такое трансляторы, компиляторы, интерпретаторы?
3) Особенности синтаксиса языка Python
4) Стиль наименования идентификаторов

5) Переменные в Python, их особенности.
6) Типы данных Python
7) Запись математических операций. Особенности.
8) Инструкция import, from.
9) Операторы присваивания, виды, особенности, обмен ссылок
10) Функция для ввода и вывода данных. Форматный вывод.
11) Расширение файла программы на языка Python.
12) Запись операторов сравнения. Оператор is.
13) Числовые эквиваленты значений True и False.
14) Функция bool().
15) Логические операторы.
16) Приоритет операций и операторов.
17) Условный оператор языка Python. Виды.
18) Инструкция pass в условном операторе.

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты
«Решение задач циклической структуры»

Задание: Составить программе по варианту для решения задач

Вычислить значение функции на отрезке с помощью операторов цикла **while** и **for.** $f = ax^3 - bx \quad a = -2 \quad b = 1.7 \quad x \in [-2;4] \quad h = 0.5$
В трёхзначном числе зачеркнули первую цифру слева; когда полученное двузначное число умножили на 7, получилось исходное трёхзначное число. Найдите это число. Выведите это число в шестнадцатеричном и восьмеричном представлении. Составить блок-схему к программе.
Вычислить сумму *n* слагаемых $Sin x + Sin^2 x + Sin^3 x + Sin^4 x + ... + Sin^n x$
Богатый дядюшка подарил мне 1 доллар в мой первый день рождения. В каждый следующий день рождения он удваивал сумму подарка и прибавлял к ней столько долларов, сколько лет мне исполнилось. Написать программу, подсчитывающую общую сумму денег, подаренных к *N*-му дню рождению.

Контрольные вопросы:
1) Оператор цикла while. Общий вид, особенности работы. Блок-схема.
2) Оператор цикла for. Общий вид, особенности работы. Блок-схема.
3) Функция range.
4) Оператор break, continue. Применение.
5) Особенности использования чисел в логическом выражении оператора while.
6) Логические значения объектов.
7) Функции ceil, floor, trunc.
8) Особенности решения задач методом перебора. Примеры задач.
9) Функции Fraction, Decimal.
10) Представление чисел в разных системах счисления.

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты
«Решение задач с использованием коллекций Python»

Задание:

1) Создать список, элементы которого содержат только согласные буквы вашего ФИО. Для этого используйте генератор списков и список, состоящий из букв вашего ФИО, полученный с помощью функции list.
2) Создайте игру «Угадай пароль». На экране пользователь видит символы, из которых может состоять пароль (укажите не более 4 символов). Затем пользователь отвечает на вопрос: «Вы будете играть в «Угадай пароль?»». При положительном ответе программа случайным образом генерирует количество (kol) символов пароля (не больше 4 символов). На экране появляется сообщение, что пароль состоит из kol символов. Пользователь вводит пароль. При правильном пароле выходит сообщение «Вы угадали пароль!». Если пароль не верен, то у пользователя спрашивают о продолжении игры. Если пользователь выходит из игры, то выводится сообщение: «Вы не смогли угадать пароль!”. Программу составляете с помощью списков. Игру можете модифицировать и сделать более интересной.
3) Создать словарь, используемый для перевода слов с русского на английский или наоборот. Не менее 10 слов. Реализовать возможность добавления новых слов в словарь, вывода списка слов в столбец, проверки наличия слова для перевода в словаре. Также можете предусмотреть другие возможности.
4) Создать словарь с помощью генератора словарей. При защите пояснить как формируется словарь.
5) Создать словарь с помощью вложенных генераторов. При защите пояснить как формируется словарь

Контрольные вопросы:

1) Что такое коллекции?
2) Способы создания строк, списков, кортежей
3) Функция range.Назначение
4) Общие методы работы с последовательностями.
5) Что такое отображения?
6) Кортеж в Python. Способы задания. Создания кортежа из одного элемента.
7) Определение позиции элемента в кортеже.
8) Привести примеры операций, функций с кортежами являющихся общими для всех последовательностей.
9) Словарь. Способы задания словаря.
10) Генераторы словарей.
11) Глубокое и поверхностное копирование в Python.

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты

«Решение задач на массивы (списки)»

Задание: Создать программы по вариантам

1. Создать массив, каждый элемент которого равен квадрату последующего индекса. Вывести полученный массив на экран. Вставьте в массив пятый элемент, значение которого равно 90. Вывести изменённый массив.
2. В массиве хранятся данные об осадках за месяц. Количество осадков получить с помощью функции *RND*. Диапазон значений выбираете самостоятельно.
   a. Определить среднемесячное количество осадков.
   b. Определить максимальное и минимальное количество осадков в году
   c. Определить количество месяцев, в которых осадков было меньше чем в мае.
   d. Сформировать массив и занести туда значения осадков в весенние и осенние месяцы.

3. Сформировать вектор, состоящий из 24 элементов. Имя вектора - первые буквы вашего имени.
   - Из вектора сформировать матрицу, состоящую из 4 строк и 6 столбцов. Имя матрицы - первые буквы вашей фамилии.
   - Определить минимальный элемент в каждом столбце матрицы.
   - Поменять местами 3 и 5 столбец матрицы
   - Вставить строку между 3 и 4 столбцом. Элементы строки – произведение индексов элементов. Вывести изменённую матрицу на экран.

Контрольные вопросы:

1) Что такое массив?
2) Как ввести элементы массива по формуле, с клавиатуры с помощью генератора псевдослучайных чисел?
3) Алгоритм сортировки массива
4) Алгоритм поиска максимального/минимального элемента в массиве
5) Алгоритм вставки строки, столбца в массив
6) Алгоритм удаления строки, столбца в массиве.

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты
«Функции пользователя в Python. Рекурсия.»

Задание: Написать программу для решения задачи с помощью функций пользователя Сформировать вектор, состоящий из 30 элементов. Имя вектора - первые буквы вашего имени.

- Из вектора сформировать матрицу, состоящую из 5 строк и 6 столбцов. Имя матрицы - первые буквы вашей фамилии.
- Определить максимальный элемент в каждом столбце матрицы.
- Поменять местами 3 и 5 строку матрицы. Вывести изменённую матрицу на экран.
- Вставить строку между 3 и 4 строками. Элементы строки – сумма индексов элементов.
- Вывести изменённую матрицу на экран.

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты
«Работа с файлами и каталогами в Python»

Задание:

В пустом каталоге создать 9 текстовых файлов. Имена файлов пользователь вводит с клавиатуры. В каждый файл генерируется случайным образом последовательность из 3 однозначных чисел. Обратите внимание, что данные в файл записываются в виде строк (преобразуйте числа в строку). Вывести содержимое каталога на экран. Вывести содержимое одного из созданных файлов, который укажет пользователь. В один из созданных файлов в конец дописать свою Фамилию, Имя и Отчество.

Контрольные вопросы:

1. Что такое файл
2. С какими файлами может осуществляться работы в Python?
3. Перечислите этапы работы с файлами.
4. Функция открытия файла.
5. Какой метод используется для закрытия файла?
6. Какие существуют режимы доступа к файлу?
7. Приведите примеры использования разных режимов доступа к файлу.
8. Как правильно задать относительный и абсолютный путь доступа к файлу?

9. Перечислите методы работы с файлами
10. Как получить список файлов и каталогов в папке?
11. Чтение данных из файла с неизвестным количеством данных?
12. Приведите пример кода, когда метод close для закрытия файла не используется

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты
«Ошибки и исключения в Python»

Задание:

1) В пустом каталоге создать 6 текстовых файлов. Имена файлов пользователь вводит с клавиатуры. В каждый файл генерируется случайным образом последовательность из 5 однозначных чисел. Обратите внимание, что данные в файл записываются в виде строк (преобразуйте числа в строку). Вывести содержимое каталога на экран. Вывести содержимое одного из созданных файлов, который укажет пользователь. Запрос на указание имени файла производить до тех пор, пока пользователь не введёт правильное имя (работа с исключением).
2) Создать программу на обработку исключения при работе с функцией sample. Для пользователя указать описание и тип ошибки. Показать работу со всеми блоками при работе с исключением.

Контрольные вопросы:

1. Что такое исключение?
2. Назовите типы ошибок в программе. Приведите примеры.
3. Назовите функции и методы, при работе с которыми возбуждаются исключения.
4. Общий вид инструкции для обработки исключения. Назначение блока else и finally. Примеры
5. Перечислите некоторые классы встроенных исключений.
6. Как работают вложенные обработчики исключений. Примеры.
7. Как получить информацию о возбуждаемом исключении в процессе выполнения программы?
8. Способы закрытия файла.
9. Менеджер контекста.

Задание на лабораторную работу

1. Получить задания в соответствии с номером своего варианта.
2. Составить схему алгоритма решения задачи.
3. Написать код программы
4. Осуществить компиляцию, отладку и тестирование программы.
5. Получить результат.
6. Оформить отчет по лабораторной работе.

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты C++

1 задание: Ввести с консоли значения переменных x, y, z. Определить значения переменных a, b, c:

$$a = \frac{2\cos(x-\pi/6)}{0{,}5+\sin^2 y} \qquad b = 1 + \frac{z^2}{3{,}2\cdot 10^{-5}+z^{2/3}} \qquad c = \mathrm{tg}\,(2{,}5x) + b\cdot\sqrt[3]{\cos ax}$$

2 Задание: Дано расстояние L в сантиметрах. Используя операцию деления нацело, найти количество полных метров в нем (1 метр = 100 см).

Контрольные вопросы:

1. Выражения какого типа используются в языке C++?
2. Какие арифметические операции используются в выражениях целого типа?
3. Какие логические операции используются в выражениях логического типа?
4. Как реализуется операция возведения в степень?

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты C++

Задание 1: Даны три переменные вещественного типа: A, B, C. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных A, B, C

Задание 2:

$y = \begin{cases} ae^{\sin x} + 2.5 \\ (\sin x) / (a + e^{x}) \\ e^{\cos x} + a \end{cases}$	$x < 0.3$ $x > 0.3$ $x = 0.3$

Контрольные вопросы:

1. Какие типы ветвлений используются в технологиях программирования?
2. Какие виды альтернативы вы знаете?
3. В каком случае используется альтернатива?
4. В каком случае используется переключатель?
5. Какие типы данных являются селекторами в переключателе?

Образец заданий для выполнения лабораторных работ и примерный перечень вопросов для их защиты C++

Задание 1: Начальный вклад в банке равен 1000 руб. Через каждый месяц размер вклада увеличивается на P процентов от имеющейся суммы (P — вещественное число, 0<P< 25). По данному P определить, через сколько месяцев размер вклада превысит 1100 руб., и вывести найденное количество месяцев K (целое число) и итоговый размер вклада S (вещественное число).

Задание 2: Вычислить значение Z при изменении X. Y - значение произвольное задается самостоятельно Результаты вывести в виде таблицы

$z = 3x + y \quad 0{,}2 \le x \le 1{,}2$ с шагом 0,2

Контрольные вопросы:

1. Какие виды циклов, управляемых условиями, используется в технологиях программирования?
2. Алгоритм работы циклов с предусловием (схема алгоритма).
3. В каких случаях циклы с предусловием превращаются в бесконечные?
4. Алгоритм работы циклов с постусловием (схема алгоритма).
5. В каких случаях циклы с постусловием превращаются в бесконечные?

6. В каких случаях циклы с предусловием не выполняются ни разу?
7. В каких случаях циклы с постусловием не выполняются ни разу?

Образец заданий для выполнения лабораторных работ C++

Задание 1: Дано вещественное число A и целое число N (> 0). Найти A в степени N:
A
N = A·A· . . . ·A
(числа A перемножаются N раз).

Задание 2: Вычислить значение Z при изменении X. Y - значение произвольное задается самостоятельно Результаты вывести в виде таблицы

$z = 3x + y \quad 0{,}2 \le x \le 1{,}2$ с шагом 0,2

Образец заданий для выполнения лабораторных работ C++

Задание 1
Строка – это правильное арифметическое выражение, содержащее только натуральные числа и знаки арифметических операций между ними (+, -, *, /). Подсчитать результат, выполняя арифметические операции в порядке их приоритета.
Входные данные:
5*3-9/3
Выходные данные:
12

Задание 2
В строке, содержащей не менее трех слов, разделенных пробелом, поменять местами второе и предпоследнее слова.
Входные данные:
abcde fghik lmnop qrst uxyz
Выходные данные:
abcde qrst lmnop fghik uxyz

Образец заданий для выполнения лабораторных работ C++

Задание 1 В целочисленном векторе vectorn найти минимальный и максимальный элементы. Предусмотреть случай, когда таких элементов нет (все элементы вектора равны между собой). Элементы вектора, расположенные между ними, отсортировать по возрастанию.
Входные данные:

```
n=10
v_min=-5
v_max=10
Исходный вектор:
7  -2  1  3  -5  5  2  0  10  1
Выходные данные:
min=-5    n_min=5
max=10    n_max=9

Выходной вектор:
7  -2  1  3  -5  0  2  5  10  1
```

Задание 2
Вывести на экран построчно элементы всех слоев целочисленного прямоугольного массива matrixnm , начиная с внешнего слоя. Вывод начинать с левого верхнего элемента слоя.

```
Входные данные для n=3, m=5:
v_min=-20
v_max=10
Исходный массив:
7    -2  -11    3   -5
5     2    0   10    1
-15   9  -12    8  -18

Выходные данные:
7  -2  -11  3  -5   1  -18  8  -12  9  -15  5
2   0   10
```

Образец заданий для выполнения лабораторных работ Golang

Задание 1: Ввести с консоли значения переменных x, y, z. Определить значения переменных **a, b, c**:

$$a = \frac{2\cos(x - \frac{\pi}{6})}{0{,}5 + sin^2 y}$$

$$b = 1 + \frac{z^2}{3{,}2 * 10^{-5} + z^{2/3}}$$

$$c = tg(2{,}5x) + b * \sqrt[3]{cosax}$$

Задание 2: Дано расстояние L в сантиметрах. Используя операцию деления нацело, найти количество полных метров в нем (1 метр = 100 см).

Образец заданий для выполнения лабораторных работ Golang

Задание 1: Дано вещественное число A и целое число N (> 0). Найти A в степени N: AN = A•A• . . . •A (числа A перемножаются N раз).

Задание 2: Вычислить значение Z при изменении X. Y - значение произвольное задается самостоятельно Результаты вывести в виде таблицы

$z = 3x + y \quad 0{,}2 \le x \le 1{,}2$ с шагом 0,2

Образец заданий для выполнения лабораторных работ Golang

Задание 1: Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить минимальное и максимальное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть

случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между минимальным и максимальным.

Задание 2: Создать текстовый файл, в который построчно записать квадратный массив massivnn, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размер массива n задать константой. Границы диапазона v_min и v_max задать с консоли. Прочитать из созданного файла k-ю строку и k-й столбец и вывести их на экран. Значение k задать с консоли. При чтении из файла считать размер массива неизвестным. Определить его при чтении файла.

Образец заданий для выполнения лабораторных работ Golang

Задание 1: Создать функцию, определяющую максимальный элемент вектора vectorn, находящийся между вторым по порядку положительным и предпоследним отрицательным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2: Создать процедуру, поворачивающую в квадратном массиве matrixnn (n – нечетное) средний наибольший ромб элементов на 90 градусов по часовой стрелке.

3.2 Типовые контрольные задания для проведения тестирования

Фонд тестовых заданий по дисциплине содержит тестовые задания, распределенные по разделам и темам, с указанием их количества и типа.

Структура фонда тестовых заданий по дисциплине

Индикатор достижения компетенции	Тема в соответствии с РПД	Характеристика ТЗ	Количество тестовых заданий, типы ТЗ
ОПК-2.1	Тема 1. Алфавит языка. Служебные слова. Константы. Переменные. Стандартные функции	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.2	Тема 2. Выражения. Операторы присваивания. Структура программы	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.3	Тема 3. Ввод и вывод данных	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.1	Тема 4. Алгоритм и его свойства. Схема алгоритма. Базовые структуры языка. Цепочка. Ветвления. Циклы.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ

		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.2	Тема 5. Структуры данных. Массивы. Строки. Множества. Записи. Файлы.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.3	Тема 6. Подпрограммы. Функции. Процедуры. Рекурсия.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-6.1 ОПК-6.2 ОПК-6.3	Тема 7. Ссылки и динамические переменные.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.1	Тема 8. Основы языков высокого уровня (Golang). Алфавит языков. Константы. Переменные. Математические функции.	Знание	4 – ОТЗ 4 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.2	Тема 9. Выражения и операторы присваивания.	Знание	4 – ОТЗ 4 – ЗТЗ
		Умение	4 – ОТЗ 4 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.3	Тема 10. Ввод и вывод данных.	Знание	4 – ОТЗ 4 – ЗТЗ
		Умение	4 – ОТЗ 4 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.1	Тема 11. Структура программы. Директивы препроцессора.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.2	Тема 12. Базовые структуры языков: цепочка, ветвления, циклы.	Знание	4 – ОТЗ 4 – ЗТЗ
		Умение	4 – ОТЗ 4 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	3 – ОТЗ 3 – ЗТЗ
ОПК-5.3	Тема 13. Структуры данных: массивы, строки, файлы.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ

		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-6.1	Тема 14. Функции	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-6.2 ОПК-6.3	Тема 15. Адреса и указатели. Указатели и массивы. Указатели и строки.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.1	Тема 16. Особенности программирования на языке Python. Среда программирования.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.2	Тема 17. Алфавит языков. Типы и модели данных. Константы. Переменные. Математические функции.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-2.3	Тема 18. Выражения и операторы присваивания.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.1	Тема 19. Базовые структуры языка: цепочка, ветвления, циклы.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.2	Тема 20. Структуры данных: списки, кортежи, массивы, строки, множества, словари, файлы.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-5.3	Тема 21. Функции. Модули.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-6.1	Тема 22. Обработка исключений.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ

		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
ОПК-6.2 ОПК-6.3	Тема 23. Классы и объекты.	Знание	2 – ОТЗ 2 – ЗТЗ
		Умение	2 – ОТЗ 2 – ЗТЗ
		Навык и (или) опыт деятельности/ действие	2 – ОТЗ 2 – ЗТЗ
		Итого	306

Полный комплект ФТЗ хранится в электронной информационно-образовательной среде ИрГУПС и обучающийся имеет возможность ознакомиться с демонстрационным вариантом ФТЗ.

Ниже приведен образец типового варианта итогового теста, предусмотренного рабочей программой дисциплины.

1. Какое значение будет выведено на экран в результате выполнения фрагмента программы?

```
x = y = z = 5
m, n = 2, 3
y *= n
print(y)
```

Ответ: 15

2. Чему равно значение выражения `-6 ** 2` в Python ?

Ответ: -36

3. Какое значение будет получено в результате выполнения кода?

```
s = 0
for x in range(2, 6):
    s += x
print (s)
```

Ответ: 14

4. Что будет выведено на экран в результате выполнения кода. Если значений несколько, то указывайте их через знак нижнего подчёркивания.
   Например, 7_8 или A_5_6

```
def my(k):
    k = copy.deepcopy(k)
    k["b"] = 5
    return k["b"]
sl = {"a":3, "b":7,"c":9}
t = my(sl)
print(t, sl["b"])
```

Ответ: 5_7

5. Дан фрагмент кода. Какие значения будут выведены на экран в результате выполнение кода?

```
b = ["Ч", "Д", "Щ", "Ъ", "Э", "Я", "Ю"]

for i in range(6):
    if i % 2 == 0 and i != 0:
        print(b[i])
```

Ответ: Щ_Э

6. Что будет выведено на экран, в результате выполнения фрагмента программы. Если выводится несколько значений, то указывайте их без пробела.

```
a = 2; b = 6; c = 1
print("{2}{1}{0}".format(a,b,c))
```

Ответ: 162

7. Какая базовая алгоритмическая структура предусматривает многократное повторение однотипных действий?

Ответ: Цикл

8. Конечный набор символов, с помощью которого могут быть образованны слова и величины языка программирования

   Ответ: **Алфавит**

9. Какой результат будет получен в результате выполнения кода. При указании ответа учитывайте синтаксис языка Python.

```
>>> a=4; b=5
>>> (a % 2 == 0) ^ (b %  2 == 0)
```

Ответ: **True**

10. Что будет выведено на экран. Если значений несколько, то вводите их БЕЗ пробела. Например, 100200

```python
class A:
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def S(self):
        return self.x*self.y
class B(A):
    def S(self):
        return 2*self.x+self.y

p1=A(3,2)
p2=B(10,5)
t1=p1.S()
t2=p2.S()
print(t1,t2)
```

Ответ: 625

11. Пользователь с клавиатуры вводит значение 4 . Что будет выведено на экран в результате выполнения фрагмента кода ?

```python
try:
    x = int(input())
    z = 3/(x-4)
except ZeroDivisionError:
    print(2)
except TypeError:
    print(5)
```

Ответ: 2

12) Какие математические выражение на языке Python записаны верно?

(модуль **math** импортирован с помощью инструкции `from math import*`**)**

A	**B**	C	D
`cos(x) ** 2`	`sin(x ** 2)`	`cos ** 2 (x)`	`|x|`

13) Будут ли являться результаты выполнения кодов одинаковыми?

```python
x = 10
while x:
    x = x - 1
    if x % 2 != 0:
        continue
    print(x, end = ' ')
```

```python
x = 10
while x:
    x = x - 1
    if x % 2 == 0:
        print(x, end = ' ')
```

A) Да B) Нет

14) Какой блок при работе с исключениями перехватывает исключения?

1) finally

2) try

3)else

4) except

5)close

15) В языке Python пока ложно условие **L** выполняется тело цикла **B**

```
While L :
    B
```

a) верно **b) не верно**

16) Перечислите виды ошибок в программах.

A	**логические**
B	статистические
C	**ошибки времени выполнения**
B	**синтаксические**
E	пунктуационные

17) Получить первую цифру четырёхзначного числа **t** позволяет операция

a) **t // 1000**
b) t div 1000
c) t % 100
d) t \\ 1000

18) Выражение **t = set ( )** создаёт

a) список
b) множество
c) словарь
d) строку
e) кортеж

19)

Представлена программа для вычисления выражения $Cos\,x + Cos^2x + Cos^3x + \cdots + Cos^nx$

```
1   from math import*
2
3   n = int(input())
4   x = float(input())
5   S = 0
6   m = 1
7   for i in range(n):
8       m = ___
9       S += m
10
11  print(S)
```

Что нужно написать в 8 строке кода вместо "_ _ _" ?

- (●) **m*cos(x)**
- ( ) **cos(x ** i)**
- ( ) **m+cos(x)**
- ( ) **m*cos(x ** i)**

20) Какие из представленных объектов являются неизменяемыми?

A	B	**C**
“привет”	[2, 3, “12”, [3, 4]]	25

21) Какая из представленных функций позволяет применять функцию к каждому элементу последовательности (или объекту, поддерживающему итерирование)?

A **map**
B filter
C zip
B reduce
E obj_iter

22) В 7 строчке кода

```
1  class Point:
2      """ Класс для представления координат точек на плоскости"""
3      x = 1
4      y = 1
5
6
7  p = Point()
```

A	B	C
создаётся экземпляр класса Point	вызывается функция пользователя Point	создаётся класс Point

23) Сопоставьте между собой фразы

A	Поверхностная копия	1	Создаёт новый составной объект и затем (по мере необходимости) вставляет в него ссылки на объекты, находящиеся в оригинале

В	Глубокая копия	2	Создаёт новый составной объект и затем вставляет в него копии объектов, находящихся в оригинале.
		3	Создаёт новый объект с указанием уровней вложения.

Ответ: А-1 В-2

3.3 Типовое задание для выполнения курсовой работы

Типовые задания выложены в электронной информационно-образовательной среде ИрГУПС, доступной обучающемуся через его личный кабинет.

Ниже приведен образец задания для выполнения курсовой работы и примерный перечень вопросов для ее защиты.

Образец типового задания для выполнения курсовой работы

1. Вычисление определителя матрицы методом Гаусса.

Вычисление обратной матрицы методом Гаусса.

2. Решение системы обыкновенных алгебраических уравнений методами
- Гаусса
- Зайделя
- по схеме Халецкого

3. Решение системы несовместных линейных уравнений методом наименьших квадратов.

4. Решение системы нелинейных и трансцендентных уравнений методом Стеффенсена.

5. Решение избыточной системы нелинейных и трансцендентных уравнений методом Ньютона.

6. Определение изолированного корня нелинейного алгебраического уравнения методами
- хорд (Ньютона-Рафсона)
- касательных
- итераций (последовательных приближений)

7. Вычисление комплексных корней алгебраического уравнения методом Ньютона-Рафсона

8. Вычисление комплексных корней алгебраического уравнения модифицированным методом Берстоу.

9. Интерполяция табличных (дискретных) функций методами

- Ньютона
- Лагранжа
- Чебышева

10. Определение характеристического полинома и собственного вектора матрицы методами
- Крылова
- Леверрье-Фаддеева

11. Приведение квадратной матрицы к форме Гессенбергера и вычисление ее собственных значений.

12. Вычисление коэффициентов характеристического полинома матрицы и обратной матрицы методом Леверрье-Фаддеева.

13. Вычисление коэффициентов характеристического полинома матрицы методом Данилевского (приведение матрицы к форме Фробениуса).

14. Разложение действительной неособенной матрицы на произведение двух треугольных (LU-разложение).

15. Аппроксимация функций, заданных аналитически, многочленами Чебышева.

16. Аппроксимация табличных (дискретных) функций методами

- наименьших квадратов (линейной функцией)
- полиномом заданной степени

17. Аппроксимация дробно-рациональной функции понижением степени ее числителя и знаменателя.

18. Аппроксимация табличных (дискретных) функций сплайн-функциями.

19. Вычисление определенного интеграла методами
- прямоугольников
- трех восьмых
- Симпсона

20. Вычисление определенного интеграла методами
- трапеций
- Ньютона-Котеса

21. Минимизация функции нескольких переменных методами
- градиента
- наискорейшего спуска

22. Минимизация функции нескольких переменных методами
- сопряженных градиентов
- Давидона-Флетчера-Пауэлла

23. Сортировка векторов методами
- простого обмена (“пузырька”)
- Хоара
- шейкерной сортировки

24. Сортировка векторов методами
- простого включения
- простого выбора
- Шелла

25. Реализация алгоритма построения LCS – самой длинной общей последовательности двух строк (алгоритма Нудельмана-Вунша)

26. Реализация алгоритма решения системы обыкновенных дифференциальных уравнений методом Рунге-Кутта четвертого порядка.

27. Вычисление корней алгебраического уравнения методами
- Ньютона-Рафсона
- Берстоу.

Образец типовых вопросов для защиты курсовых работ

1. Методы сортировки векторов
2. Метод Гауса
3. Парадигмы программирования.
4. Стили программирования
5. Математические операции
6. Алгоритм и его свойства. Блок-схемы алгоритмов.
7. Операторы переходов
8. Операторы управления вычислительным процессом. Условные операторы условия if.
9. Программа, функция, оператор. Константы и переменные
10. Массивы. Элементарные операции над массивами.
11. Структура программы. Линейная программа.
12. Функции. Объявление, инициализация и вызов функции

3.4 Перечень теоретических вопросов к зачету

(для оценки знаний)

1. Константы (целые, вещественные, логические, символьные, строковые). Переменные, идентификаторы.
2. Стандартные функции.

3. Выражения целого, вещественного и логического типов. Операторы присваивания.
4. Структуры данных – массивы. Векторы и матрицы. Многомерные матрицы.
5. Ветвления: альтернатива и переключатель. Блок-схемы, реализация. Часто встречающиеся ошибки при программировании ветвлений.
6. Циклы с предусловием: блок-схема, реализация. Часто встречающиеся ошибки при программировании циклов с предусловием.
7. Циклы с постусловием: блок-схема, реализация. Часто встречающиеся ошибки при программировании циклов с постусловием.
8. Циклы с параметром: блок-схема, реализация. Часто встречающиеся ошибки при программировании циклов с параметром.
9. Структуры данных – массивы.
10. Подпрограммы-функции. Формальные и фактические параметры. Локальные и глобальные переменные.
11. Подпрограммы-процедуры. Рекурсия
12. Программные модули. Структура модуля.
13. Ссылки и динамические переменные
14. Динамические структуры данных. Связные списки. Формирование связного списка.
15. Сортированные списки. Формирование сортированного списка.
16. Бинарные деревья. Сортировка на бинарных деревьях. Создание бинарного дерева
17. Поиск, добавление и удаление узла из бинарного дерева
18. Жизненный цикл программы. Модели жизненного цикла
19. Стили программирования
20. Отладка программы. Методы и средства отладки
21. Тестирование программы. Методы тестирования
22. Структурное и функциональное тестирование.

3.5 Перечень типовых простых практических заданий к зачету

(для оценки умений)

Разработать алгоритм решения задачи, написать код программы на языке C++,Golang, отладить программу и протестировать ее:

1. Найти все натуральные трехзначные числа, в записи которых нет одинаковых цифр: ***102,103,…,987***.
2. Найти все натуральные трехзначные числа, крайние цифры в которых симметричны относительно средней: ***101,111, 121,…,999***.
3. Найти все седловые точки целочисленного массива A_{nn}. ***Седловая*** точка – это элемент массива, являющийся одновременно наименьшим в своей строке и наибольшим в своем столбце.
4. Найти все натуральные трехзначные числа, сумма цифр которых равна заданному числу ***N.***
5. Массив A_{nm} заполнен построчно слева направо и сверху вниз натуральными числами от ***1*** до ***n*m***. По заданному значению элемента массива определить его индексы – номер строки и столбца.
6. Сдвинуть на шаг вправо элементы заданного вектора A_n, причем последний его элемент должен занять первое место.
7. Из записи заданного натурального числа ***N*** удалить цифру, стоящую на заданном месте ***M.***
8. В заданном целочисленном массиве A_{nm} поменять местами минимальный и максимальный элементы массива.
9. В заданном целочисленном массиве A_{nm} поменять местами первый отрицательный и последний положительный элементы.
10. В заданном целочисленном векторе A_n поменять местами первый отрицательный и последний положительный элементы.
11. В заданном целочисленном векторе A_n поменять местами максимальный и минимальный элементы вектора.

12. В заданном целочисленном массиве A_{nn} поменять местами главную и побочную диагональ.
13. Произвести сквозную сортировку по возрастанию элементов заданного целочисленного массива A_{nm}.
14. Задан символьный массив A_{nn} со случайно расставленными, не идущими подряд строчными буквами латинского алфавита. Расставить их по алфавиту.
15. В заданном целочисленном векторе A_n расположить элементы так, чтобы первыми шли положительные элементы, далее – отрицательные, далее – нули, причем внутри каждой группы прежний порядок их следования должен сохраниться.
16. В заданном целочисленном векторе A_n переместить все нулевые элементы в его начало, сохраняя порядок следования остальных элементов.
17. В заданном целочисленном массиве A_{nn} поменять местами строку с наименьшей суммой элементов и столбец с наибольшей.
18. Массив A_{nm} пронумерован построчно слева направо и сверху вниз натуральными числами от 1 до $n*m$. По заданному номеру клетки определить номера всех клеток, соприкасающихся с ней углами.
19. Перевернуть любое введенное натуральное число и определить сумму его цифр.
20. Найти все трехзначные натуральные числа, десятичная запись которых есть возрастающая последовательность цифр: ***123, 124, …,789.***
21. В заданном целочисленном массиве A_{nn} определить сумму элементов главной и побочной диагоналей.
22. В заданном целочисленном массиве A_{nn} перевернуть главную диагональ (поплавок).
23. В заданном целочисленном массиве A_{nn} перевернуть побочную диагональ (поплавок).
24. Сформировать вектор B_n из элементов заданной целочисленной матрицы A_{nn} по следующему алгоритму: очередной элемент вектора b_i – это максимальный элемент ***i-ой*** строки исходной матрицы.
25. Сформировать вектор B_n из элементов заданной целочисленной матрицы A_{nn} по следующему алгоритму: очередной элемент вектора b_i – это минимальный элемент ***i-ого*** столбца исходной матрицы.
26. Сформировать вектор B_n из элементов заданной целочисленной матрицы A_{nn} по следующему алгоритму: сначала записать в вектор все положительные элементы матрицы в порядке их следования, затем – все отрицательные, в конце – все нули.
27. В заданном целочисленном массиве A_{nn} определить максимальный элемент из элементов, расположенных выше главной диагонали, и минимальный – ниже главной диагонали.
28. В заданном целочисленном массиве A_{nn} определить максимальный элемент из элементов, расположенных выше главной и побочной диагоналей.

3.6 Перечень типовых практических заданий к зачету

(для оценки навыков и (или) опыта деятельности)

1. Вывести на экран текст "Silence is golden".
2. Вывести на экран текущее название дня недели, название месяца и свое имя. Каждое слово должно быть в отдельной строке.
3. Вывести на экран пять строк из нулей, причем количество нулей в каждой строке равно номеру строки.
4. Вывести на экран прямоугольник, заполненный буквами А. Количество строк в прямоугольнике равно 5, количество столбцов равно 8.
5. Вывести на экран букву "W" из символов "*" (звездочка).
6. Пользователь вводит два числа. Найдите сумму и произведение данных чисел.
7. Пользователь вводит число. Выведите на экран квадрат этого числа, куб этого числа.
8. Пользователь вводит три числа. Увеличьте первое число в два раза, второе числа уменьшите на 3, третье число возведите в квадрат и затем найдите сумму новых трех чисел.

9. Пользователь вводит три числа. Найдите среднее арифметическое этих чисел, а также разность удвоенной суммы первого и третьего чисел и утроенного второго числа.

10. Пользователь вводит сторону квадрата. Найдите периметр и площадь квадрата.

11. Дано число. Если оно больше 3, то увеличить число на 10, иначе уменьшить на 10.

12. Дано число. Если оно меньше 7, то вывести Yes, если больше 10, то вывести No, если равно 9, то вывести Error.

13. Пользователь вводит номер месяца, вывести название месяца.

14. Дано два числа. Вывести наибольшее из них.

15. Дано два числа. Вывести yes, если они отличаются на 100, иначе вывести No.

16. Даны два числа. Если первое число больше второго, то вывести yes, иначе поменять значения этих переменных и вывести их на экран.

17. Пользователь вводит англ. букву, вывести следующие три по алфавиту. Если алфавит закончился, то вывести циклично с начала алфавита, то есть если z, то a b c. Вывод только маленьких букв. Учесть, что пользователь может ввести заглавную

18. Вывести англ. алфавит по 5 букв в строке.

19. Вывести квадрат 7 на 7 из случайных букв. Материал сайта www.itmathrepetitor.ru

20. Пользователь вводит положительное целое число. Зашифровать каждую цифру серией из букв (конкретный принцип составления серии букв разработать самостоятельно).

21. Создать матрицу 3 x 4, заполнить ее числами 0 и 1 так, чтобы в одной строке была ровно одна единица, и вывести на экран.

22. Создать и вывести на экран матрицу 2 x 3, заполненную случайными числами из [0, 9].

23. Дана матрица. Вывести на экран первый и последний столбцы.

24. Дана матрица. Вывести на экран первую и последнюю строки.

25. Дана матрица. Вывести на экран все четные строки, то есть с четными номерами.

26. Создайте структуру с именем student, содержащую поля: фамилия и инициалы, номер группы, успеваемость (массив из пяти элементов). Создать массив из десяти элементов такого типа, упорядочить записи по возрастанию среднего балла. Добавить возможность вывода фамилий и номеров групп студентов, имеющих оценки, равные только 4 или 5.

27. Создайте структуру с именем train, содержащую поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа train, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массив по пункту назначения, причем поезда с одинаковыми пунктами назначения должны быть упорядочены по времени отправления.

28. Дан текстовый файл, содержащий целые числа. Удалить из него все четные числа.

29. В данном текстовом файле удалить все слова, которые содержат хотя бы одну цифру.

30. Дан текстовый файл. Создать новый файл, каждая строка которого получается из соответствующей строки исходного файла перестановкой слов в обратном порядке.

31. Дан текстовый файл. Создать новый файл, состоящий из тех строк исходного файла, из чисел которых можно составить арифметическую прогрессию.

3.7 Перечень теоретических вопросов к экзамену

(для оценки знаний)

1. Алгоритм. Свойства алгоритма. Блоки для создания блок-схем алгоритма.
2. Языки программирования. Классификация языков программирования.
3. Языки программирования по различным классификациям. Примеры.

4. Парадигмы программирования.
5. Трансляторы, интерпретаторы, компиляторы.
6. Интегрированные среды разработки (IDE).
7. Язык Python и его особенности. IDE для написания программ на Python.
8. Особенности синтаксиса языка Python.
9. Математические операции на языке Python (сложение, умножение, деление, возведение в степень, целочисленное деление, нахождение остатка от деления) и их особенности.
10. Типы данных в Python. Определение типа объекта в Python.
11. Переменные в Python и их особенности.
12. Создание комментариев в Python.
13. Идентификаторы. Соглашение о наименовании.
14. Способы задания начальных значений для переменных.
15. Функция ***input***. Синтаксис, назначение. Особенности использования.
16. Оператор присвоения. Виды оператора присвоения.
17. Обмен значение переменных (ссылок).
18. Импортирование библиотек и модулей. Инструкция ***import***, ***from***. Особенности использования.
19. Функция ***print***. Параметры ***sep***, ***end***. Метод ***format***. Особенности вывода целых, вещественных чисел.
20. Сокращённая форма записи оператора присвоения.
21. Операторы сравнения языка Python.
22. Логические значения ***True*** и ***False***, числовые эквиваленты.
23. Функция для определения логического значения объекта.
24. Какие объекты возвращают значение ***True*** и ***False***.
25. Оператор ***is***. Примеры использования.
26. Логические операторы Python. Примеры записи.
27. Приоритет логический операций и операторов.
28. Условный оператор Python. Виды, блок-схемы.
29. Оператор ***pass***. Назначение.
30. Тип данных ***float*** и его особенности. Модуль ***fractions***, ***decimal***.
31. Оператор цикла **while** в Python. Общий вид, блок-схема.
32. Реализация бесконечно цикла с помощью оператора ***while.***
33. Оператор ***break***. Назначение. Примеры использования.
34. Оператор ***continue***. Примеры использования.
35. Функция ***range***. Назначение. Особенности использования.
36. Оператор цикла **for.** Общий вид, блок-схема.
37. Нахождение суммы/произведения с помощью операторов цикла **while** и **for.**
38. Функции модуля **math**: ***ceil, floor, trunc***.
39. Вычисление значений функций на отрезке с помощью операторов цикла **while** и **for.**
40. Примеры задач, решаемых методом перебора.
41. Последовательности Python. Общие операции, функции и методы, применяемые ко всем последовательностям. Примеры.
42. Понятие среза.
43. Строки в Python. Индексация элементов строки. Базовые операции со строками.
44. Функции для работы с символами.
45. Методы работы со строками. Понятие методы. Привести примеры методов.

46. Списки в Python. Способы создания списков. Синтаксис генератора списков. Примеры использования генератора списков.
47. Генерация вложенных списков.
48. Обращение к элементу списка. Обращение к элементам вложенных списков.
49. Применение общих операций и методов для работы с последовательностями к спискам.
50. Индексация. Срезы.
51. Методы работы со списками (привести примеры нескольких методов)
52. Добавление элемента в список, удаление элемента из списка.
53. Кортеж. Способы создания кортежей.
54. Словари. Правила при использовании словарей.
55. Способы создания словаря.
56. Генераторы словарей. Вложенные генераторы словарей.
57. Изменяемые и неизменяемые объекты. Глубокое и поверхностное копирование.
58. Функции получения псевдослучайных чисел.
59. Применение списков для работы с массивами одномерными и двумерными.
60. Обращение к элементам двумерного массива.
61. Задание элементов одномерного и двумерного массива.
62. Способы ввода элементов массива.
63. Вывод элементов двумерного массива в виде матрицы (2 способа).
64. Создание копий строки/столбца в двумерном массиве.
65. Реализация алгоритмов работы с одномерным массивом на языке Python:
    a. Ввод /вывод элементов
    b. Обмен местами элементов
    c. Сортировка элементов
    d. Вставка/удаление элемента
    e. Определение суммы/произведения.
    f. Определение суммы/произведения/количества элементов, удовлетворяющих условию.
    g. Формирование вектора из элементов, удовлетворяющих условию.
    h. Поиск максимального/минимального элемента и его номера
66. Реализация алгоритмов работы с двумерным массивом на языке Python:
    a. Поиск максимального/минимального элемента в двумерном массиве и его индексов
    b. Определение суммы/произведения элементов массива.
    c. Определение количества/произведения/суммы элементов двумерного массива, удовлетворяющих заданному условию.
    d. Определение количества/суммы/произведения/максимального/минимального элемента в строках/столбцах двумерного массива.
    e. Формирование одномерного массива из элементов двумерного массива.
    f. Формирование двумерного массива из элементов вектора
    g. Обмен местами строк/столбцов двумерного массива
    h. Сортировка элементов по строкам/столбцам в двумерном массиве
    i. Удаление/вставка строки/столбца в двумерном массиве.
67. Подпрограммы Python. Определение. Назначение.
68. Задание функции пользователя. Синтаксис описания функции. Оператора ***pass.***
69. Функции без параметров и с параметрами. Формальные и фактические параметры.

70. Расположение описаний функций.
71. Использование необязательных параметров в функции. Сопоставление параметров по ключам.
72. Особенности передачи значений в функцию из списка, кортежа, словаря.
73. Особенности задания функции с переменным числом параметров.
74. Особенности передачи объектов неизменяемого и изменяемого типа в функцию.
75. Анонимные функции (лямбда-функции).
76. Модули и пакеты в Python.
77. Создание модулей пользователя в Python.
78. Создание документации по функциям пользователя.
79. Работа с файлами в Python. Понятие файла. Этапы работы с файлами.
80. Функция ***open***. Метод ***close.***
81. Особенности режимов доступа к файлу.
82. Методы работы с файлами: write, writelines , read, readline, readlines, seek.
83. Чтение файла с неизвестным количеством данных.
84. Ошибки и исключения в Python
85. Основы объектно-ориентированного программирования.

3.8 Перечень типовых простых практических заданий к экзамену

(для оценки умений)

1. Создайте блок-схему решения задачи: Вычислить значение функции на отрезке:
   $f = ax^3 - bx \qquad a = -2 \qquad b = 1.7 \qquad x \in [-2;4] \quad h = 0.5$

2. Напишите программу для:
   - определения последней цифры трёхзначного числа;
   - определения второй цифры четырёхзначного числа;
   - определения среднего арифметического чисел, введённых с клавиатуры;
3. Запишите с помощью одного оператора присвоения, что три переменные x, y, z являются ссылками на объект целого типа 5.

4. x=5, y=6. Используя один оператор присвоения, обменяйте значения переменных x и y.

5. Напишите программу нахождения суммы первых 10 натуральных чисел.

6. Создайте словарь, в котором будет находиться информация о вашем возрасте, имени и фамилии.

7. Что означает запись **a[0][3]** для списка с именем **a**?

8. Запишите сокращённую форму для оператора присвоения.

9. Вычислите значение выражения $\sqrt[3]{\frac{x^5+7}{|-6|\cdot y}}$.

10. Дан фрагмент программы

```
s = 0
i = 2
while i <= 5:
    s = s + i
    i += 1
print (s)
```

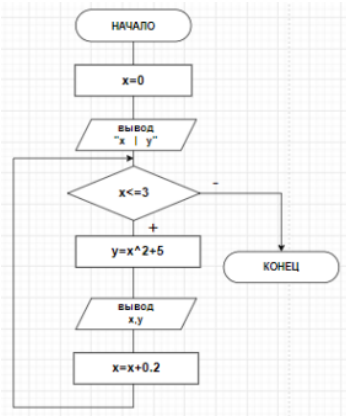
- сколько раз будет выполняться тело цикла?
- какое значение будет выведено на экран?

11. Что будет являться результатом выполнения фрагмента программы

```
i = 100
while i:
    print(i)
    i -= 1
```

?

12. Создайте программу на языке Python по заданной блок-схеме.



3.9 Перечень типовых практических заданий к экзамену

(для оценки навыков и (или) опыта деятельности)

1. Вычислить значение функции на отрезке с помощью операторов цикла **while** и **for.** Значение функции вывести в текстовый файл.

$$f = ax^3 - bx \qquad a = -2 \qquad b = 1.7 \qquad x \in [-2;4] \quad h = 0.5$$

2. В трёхзначном числе зачеркнули первую цифру слева; когда полученное двузначное число умножили на 7, получилось исходное трёхзначное число. Найдите это число. Выведите это число в шестнадцатеричном и восьмеричном представлении. Составить блок-схему к программе.

3. Вычислить сумму *n* слагаемых $Sin\,x + Sin^2 x + Sin^3 x + Sin^4 x + ... + Sin^n x$.

4. Шахматная доска содержит 8 × 8 клеток. Горизонтали и вертикали обозначаются цифрами от 1 – 8. Написать программу, проверяющую, можно ли с первой заданной клетки попасть на вторую одним ходом ладьи.

5. С клавиатуры ввести путь доступа к файлу. В путь доступа к файлу входят не менее 5 папок. Вывести только имена папок на экран в виде списка.

6. Известен первый элемент массива A: A[1]=6. Создать массив, каждый элемент которого равен квадрату предыдущего индекса. Вывести полученный массив. Удалить третий элемент в массиве. Вывести изменённый массив.

10. В массиве хранятся данные, соответствующие росту учеников класса. Значение роста получить с помощью функции *RND*. Диапазон значений для роста укажите самостоятельно.
    b. Определить средний рост учеников в классе.
    c. Определить рост самого низкого ученика в классе.
    d. Определить сколько учеников отберут в баскетбольную секцию, если туда берут с ростом выше 170 см.
    e. Вывести рост учеников класса в порядке возрастания.
11. Создать модуль пользователя с функциями для вычисления элементов треугольника.
12. Используя генератор списков создать вектор, состоящий из чётных чисел, начиная с числа n. Сформировать матрицу из элементов вектора. Определить сумму элементов каждого столбца.
13. Создать матрицу. Определить произведение элементов, значения которых меньше Z, в строках матрицы. Заменить все элементы в чётных строках на значение суммы всех элементов матрицы. Вывести значение суммы всех элементов в файл.
14. Сформировать матрицу. Из элементов матрицы, значения которых больше заданного числа сформировать вектор. Найти произведение нечётных элементов столбцов матрицы.
15. Привести пример программы с обработкой исключения.

4. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности

В таблице приведены описания процедур проведения контрольно-оценочных мероприятий и процедур оценивания результатов обучения с помощью оценочных средств в соответствии с рабочей программой дисциплины.

Наименование оценочного средства	Описания процедуры проведения контрольно-оценочного мероприятия и процедуры оценивания результатов обучения
Лабораторная работа	Защита лабораторных работ проводится во время лабораторных занятий. Во время проведения защиты лабораторной работы пользоваться учебниками, справочниками, конспектами лекций, тетрадями не разрешено. Преподаватель на лабораторной работе, предшествующей занятию проведения защиты лабораторной работы, доводит до обучающихся: номер защищаемой лабораторной работы, время на защиту лабораторной работы. Преподаватель информирует обучающихся о результатах защиты лабораторной работы сразу после ее контрольно-оценочного мероприятия
Курсовая работа	Ход выполнения разделов курсовой работы в рамках текущего контроля оценивается преподавателем исходя из объемов выполненных работ в соответствие со шкалами оценивания. Преподаватель информирует обучающихся о результатах оценивания выполнения курсового проекта сразу после контрольно-оценочного мероприятия. В ходе защиты курсовой работы обучающийся делает доклад протяженностью 5 – 7 минут. Преподаватель ставит окончательную оценку за курсовую работу после завершения защиты, учитывая уровень ее защиты

Для организации и проведения промежуточной аттестации составляются типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.

Перечень теоретических вопросов и типовые практические задания разного уровня сложности для проведения промежуточной аттестации обучающиеся получают в начале семестра через электронную информационно-образовательную среду ИрГУПС (личный кабинет обучающегося).

Описание процедур проведения промежуточной аттестации в форме зачета и оценивания результатов обучения

При проведении промежуточной аттестации в форме зачета преподаватель может воспользоваться результатами текущего контроля успеваемости в течение семестра. С целью использования результатов текущего контроля успеваемости, преподаватель подсчитывает среднюю оценку уровня сформированности компетенций обучающегося (сумма оценок, полученных обучающимся, делится на число оценок).

Шкала и критерии оценивания уровня сформированности компетенций в результате изучения дисциплины при проведении промежуточной аттестации в форме зачета по результатам текущего контроля (без дополнительного аттестационного испытания)

Средняя оценка уровня сформированности компетенций по результатам текущего контроля	Шкала оценивания
Оценка не менее 3,0 и нет ни одной неудовлетворительной оценки по текущему контролю	«зачтено»
Оценка менее 3,0 или получена хотя бы одна неудовлетворительная оценка по текущему контролю	«не зачтено»

Если оценка уровня сформированности компетенций обучающегося не соответствует критериям получения зачета без дополнительного аттестационного испытания, то

промежуточная аттестация проводится в форме собеседования по перечню теоретических вопросов и типовых практических задач или в форме компьютерного тестирования.

Промежуточная аттестация в форме зачета с проведением аттестационного испытания проходит на последнем занятии по дисциплине.

При проведении промежуточной аттестации в форме компьютерного тестирования вариант тестового задания формируется из фонда тестовых заданий по дисциплине случайным образом, но с условием: 50 % заданий должны быть заданиями открытого типа и 50 % заданий – закрытого типа.

Описание процедур проведения промежуточной аттестации в форме экзамена и оценивания результатов обучения

Промежуточная аттестация в форме экзамена проводится путем устного собеседования по билетам или в форме компьютерного тестирования.

При проведении промежуточной аттестации в форме собеседования билеты составляются таким образом, чтобы каждый из них включал в себя теоретические вопросы и практические задания.

Билет содержит: два теоретических вопроса для оценки знаний. Теоретические вопросы выбираются из перечня вопросов к экзамену; два практических задания: одно из них для оценки умений (выбирается из перечня типовых простых практических заданий к экзамену); другое практическое задание для оценки навыков и (или) опыта деятельности (выбираются из перечня типовых практических заданий к экзамену).

Распределение теоретических вопросов и практических заданий по экзаменационным билетам находится в закрытом для обучающихся доступе. Разработанный комплект билетов (25-30 билетов) не выставляется в электронную информационно-образовательную среду ИрГУПС, а хранится на кафедре-разработчике фондов оценочных средств.

На экзамене обучающийся берет билет, для подготовки ответа на экзаменационный билет обучающемуся отводится время в пределах 45 минут. В процессе ответа обучающегося на вопросы и задания билета, преподаватель может задавать дополнительные вопросы.

Каждый вопрос/задание билета оценивается по четырехбалльной системе, а далее вычисляется среднее арифметическое оценок, полученных за каждый вопрос/задание. Среднее арифметическое оценок округляется до целого по правилам округления

При проведении промежуточной аттестации в форме компьютерного тестирования вариант тестового задания формируется из фонда тестовых заданий по дисциплине случайным образом, но с условием: 50 % заданий должны быть заданиями открытого типа и 50 % заданий – закрытого типа.

Образец экзаменационного билета

ИРГУПС 20__-20__ учебный год	**Экзаменационный билет № 1** **по дисциплине «Программирование»**	Утверждаю: Заведующий кафедрой «_____» ИрГУПС _____
1.Выражения и операторы присваивания 2. Циклы с параметром 3. Найти в целочисленном квадратном массиве минимальный и максимальный элементы и поменять их местами 4. В заданной строке символов определить, каких гласных букв там нет		