

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлутков
Должность: директор
Дата подписания: 16.06.2026 21:58:02
Уникальный программный ключ:
880f7c07c583b07b775f6604a630281b13ca9fd2

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.09 Алгоритмизация и программирование
(индекс, наименование дисциплины в соответствии с учебным планом)

38.03.01 Экономика
(код, наименование направления подготовки/специальности)

Мировая экономика
(наименование образовательной программы)

Очная форма обучения
(форма обучения)

Год набора - 2026

Санкт-Петербург

Автор(ы)-составитель(и) РПД:

Рассказов Владимир Александрович, кандидат технических наук, доцент
кафедры бизнес-информатики

Заведующий кафедрой:

Наумов Владимир Николаевич, доктор военных наук, профессор

Рабочая программа дисциплины Б1.О.09 Алгоритмизация и программирование
одобрена на заседании кафедры экономики факультета экономики и финансов
СЗИУ РАНХиГС.

протокол № 10 от «26» марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.
2. Объем и место дисциплины в структуре образовательной программы.
3. Содержание и структура дисциплины.
4. Типы оценочных материалов, показатели и критерии их оценивания.
5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам.
6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине.
7. Методические материалы по освоению дисциплины.
8. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет».
9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.О.09 Алгоритмизация и программирование обеспечивает формирование у обучающихся следующих универсальных, общепрофессиональных и профессиональных компетенций:

ОТФ/ТФ и реквизиты ПС	Код компетенции	Наименование компетенции	Код индикатора достижения компетенций	Наименование индикатора достижения компетенций	Образовательный результат
	ОПК ОС-6	Способен использовать современные информационные технологии и программные средства при решении профессиональных задач	ОПК ОС-6.1	Реализует основные типовые алгоритмы решения задач на языке программирования высокого уровня	З-1. Знает основные типовые алгоритмы решения задач на языке программирования высокого уровня. У-1. Умеет разрабатывать основные типовые алгоритмы решения задач на языке программирования высокого уровня.
			ОПК ОС-6.2	Применяет языки программирования при решении профессиональных задач	З-1. Знает языки программирования при решении профессиональных задач. У-1. Умеет применять языки программирования при решении профессиональных задач.

2. Объем и место дисциплины в структуре ОП ВО

Объем дисциплины

Общий объем дисциплины – 10 з.е., 360 ак. ч.

Контактная работа обучающихся с преподавателем по видам учебных занятий: 150 ак.ч. на контактную работу с преподавателем, из них 64 ак.ч. лекций и 64 ак.ч семинарских занятий, 4 ак.ч. на консультацию к экзамену, 18 ак.ч. на аттестацию в период экзаменационных сессий. На самостоятельную работу обучающихся на самостоятельную работу обучающихся – 174 ак.ч., контроль – 36 ак.ч.

Дисциплина Б1.О.06 «Алгоритмизация и программирование» изучается на 1 курсе в 1 семестре и 2 семестре студентами очной формы обучения.

Дисциплина Б1.О.06 «Алгоритмизация и программирование» опирается на знание общеобразовательных дисциплин Б1.О.05 «Математический анализ», Б1.О.06 «Алгебра».

Дисциплина Б1.О.06 «Алгоритмизация и программирование» создаёт необходимые предпосылки для освоения программ таких дисциплин, как Б1.В.14 «Информатика» и ряда дисциплин по выбору студента.

	сетей.												
Промежуточная аттестация (2 семестр)	29							2	9		18		Экзамен
<i>Итого 2 семестр</i>	180	32				32		2	9		18	87	
Итого	360	64				64		4	18		36	174	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).

ИК – индивидуальные консультации.

КР – контрольная работа

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения

СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.

СРэк – самостоятельная работа на подготовку к экзамену.

СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

Т – тестирование.

ПКЗ - практическое контрольное задание

3.2 Содержание дисциплины

Тема 1. Информационные процессы. Основы алгоритмизации. ОПК ОС-6.1.

Определение и свойства алгоритма. Способы записи и типы алгоритмов. Технические средства реализации информационных систем. Программные средства реализации информационных процессов. Парадигмы программирования.

Технологии создания программ. Методика разработки простых программ решения отдельных задач. Понятие и этапы жизненного цикла разработки ПО. Аттестация, верификация и тестирование программного кода. Модели жизненного цикла ПО. Экстремальное программирование *XP*.

Основы языка программирования высокого уровня. Создание переменных. Операции с переменными. Вывод переменных. Пользовательский ввод. Ошибки и исключения. Описание стандартных типов данных. Проверка типа данных. Явное и неявное преобразование типов. Арифметические операции. Порядок операций. Операторы сравнения. Логические операции. Операции со строками. Инструкции *IF*, *IF-ELIF-ELSE*, вложенные инструкции *IF* (*IF* внутри *IF*). Объявление функции. Циклы. Цикл *for*, *while*.

Вычислительная сложности алгоритмов. Асимптотический анализ трудоемкости алгоритмов. Оценка сложности алгоритмических конструкций. Типовые функции временной оценки сложности алгоритмов. Составные классы сложности.

Основы ООП. Классы и объекты. Атрибуты класса. Функции внутри класса и объекта. Инициализация объекта. Наследование. Полиморфизм в операциях.

Тема 2. Структуры и типы данных. Реализация базовых алгоритмов. ОПК ОС-6.1.

Концепция абстрактных типов данных. Абстрактные типы данных «Вектор», «Список», «Стек», «Очередь», «Дек».

Основные типы данных языка программирования. Определение и основные характеристики списков, кортежей. Определение и основные характеристики матриц. Стандартные сортировки языка программирования. Множества. Операции с множествами.

Структуры данных для ассоциированных списков. Хеш-функции, хеш-суммы, хеш-таблица. Коллизии. Алгоритм открытой адресации.

Алгоритм раздельного хранения цепочек в списках. Расширяемые хеш-таблицы.

Словари в языке программирования, требования к ключу словаря. Операции над словарями. Методы словаря.

Определение порядка и сортировки. Алгоритмы глупой сортировки и их асимптотика. Пузырьковая сортировка. Сортировка выбором. Сортировка вставками. Алгоритмы эффективной сортировки и их асимптотика. Сортировка слиянием. Быстрая сортировка. Пирамидальная сортировка. *Tim Sort*.

Алгоритмы поиска. Линейный поиск. Бинарный поиск. Левый и правый бинарные поиски. Задача поиска двух элементов с фиксированной суммой. *Jump Search*. Поиск Фибоначчи. Экспоненциальный поиск. Интерполяционный поиск.

Основы динамического программирования. Задача о кузнечике. Алгоритмы поиска чисел Фибоначчи. Задача о роботе. Задача о роботе с препятствиями. Алгоритм поиска простых чисел: наивный алгоритм и решето Эратосфена. Алгоритм поиска НОД. Алгоритм поиска НОК.

Поиск символа и словарь со счетчиком. Вхождение подмножества символов (поиск подстроки). Проверка на палиндром. Префикс-функция и алгоритм Кнута–Морриса–Пратта.

Тема 3 – Основы *Data Science*. Использование языка программирования для решения экономических задач. ОПК ОС-6.2.

Библиотеки языка программирования для работы с многомерными массивами, обработки и анализа данных. Атрибуты массивов. Индексация массива: доступ к отдельным элементам. Срезы массивов: доступ к подмассивам. Создание копий массивов. Слияние, разбиение, изменение формы массивов. Сортировка массивов. Операции над массивами. Фильтрация данных.

Агрегирование: минимум, максимум, суммирование значений в массиве. Многомерные сводные показатели. Описательная статистика. Корреляция и ковариация.

Доступ к файловым данным в языке программирования. Чтение и запись текстовых данных. Чтение и запись файлов с табличными данными.

Библиотека визуализации данных языка программирования. Линейные диаграммы. Визуализация погрешностей. Столбчатые диаграммы. Круговые диаграммы. Гистограммы. Настройка графика: оси, метки, цвета и стили линий.

Введение в машинное обучение. Обучение и оценка модели. Регрессионный анализ и классификация. Создание прогнозов. Алгоритмы кластеризации. Плоская и иерархическая кластеризация. Снижение размерности признакового пространства. Метод главных компонент (*PCA*).

Регулярные и нерегулярные временные ряды. Общие методы анализа временных рядов. Вычисление процентных изменений. Вычисление скользящего окна. Обработка временных рядов, анализ зависимости между переменными.

Тема 4. Графовые алгоритмы и анализ экономических сетей. ОПК ОС-6.2.

Понятие графа. Ориентированные и неориентированные графы. Способы описания графов. Библиотека *NetworkX*. Создание графов, доступ к ребрам и узлам. Добавление атрибутов к графам, узлам и ребрам. Операции с графами. Отображение графов.

Волновой алгоритм. Поиск в глубину. Определение компонентов связности. Раскраска графов. Алгоритм Прима и Краскала. Алгоритмы нахождения минимального остовного дерева (*MST*). Алгоритм Прима. Алгоритм Краскала.

Алгоритмы нахождения кратчайшего пути в графе. Основные принципы алгоритма Дейкстры. Алгоритм Беллмана – Форда. Алгоритм Флойда – Уоршелла.

Задача коммивояжера: методы, алгоритмы. Полный перебор. Алгоритм Беллмана-Хелда-Карпа. Жадный алгоритм. Муравьиный алгоритм (Эвристический).

Поиск максимального потока в сети. Транспортные сети и потоки. Метод Форда-Фалкерсона. Остаточные сети. Увеличивающие пути. Разрезы транспортных сетей. Базовый алгоритм Форда-Фалкерсона.

4. Типы оценочных материалов, показатели и критерии оценивания

4.1 Оценочные материалы по дисциплине Б1.О.06 Алгоритмизация и программирование входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа — это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа — это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4 Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В). 	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4). 	Ответ считается верным, если правильно указаны цифры или буквы

<p>Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных</p>	<p>Прочитайте текст, выберите правильные ответы</p>	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранных вариантов ответа (например, 1 4 или А Г). 	<p>Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)</p>
<p>Задание закрытого типа на установление последовательности</p>	<p>Прочитайте текст и установите последовательность</p>	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БА или 135). 	<p>Ответ считается верным, если правильно указана вся последовательность цифр</p>
<p>Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием</p>	<p>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа</p>	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) 	<p>Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа</p>

выбора		выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).	
Задание открытого типа с развернутым ответом	Прочитайте текст и запишите развернутый обоснованный ответ	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ 	<p>Ответ считается верным:</p> <ol style="list-style-type: none"> 1. Отсутствие фактических ошибок. 2. Раскрытие объема используемых понятий (полнота ответа). 3. Обоснованность ответа (наличие аргументов). 4. Логическая последовательность излагаемого материала.

4.5 Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Итоговая балльная оценка	Традиционная система	Бинарная система	ECTS	
			Для традиционной системы	Для бинарной системы
95-100	Отлично	Зачтено	A	P/ Passed
85-94			B	P/ Passed
75-84	Хорошо		C	P/ Passed
65-74			D	P/ Passed
55-64	Удовлетворительно		E	P/ Passed
0-54	Неудовлетворительно	Не зачтено	F	F/Failed

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
80 баллов	20 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

Тестирование (Т), практические контрольные задания (ПКЗ), контрольная работа (КР).

Тема 1 - Информационные процессы, основы алгоритмизации. ОПК ОС-6.1.

Тест 1.1.

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.

2. Внимательно прочитать предложенные варианты-ты ответа.

3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).

- 1) Что такое информационный процесс?
 - a) Любые действия, заключающиеся в получении, создании, сборе, передаче, хранении и обработке информации.
 - b) Действия по фиксации получаемой информации с помощью специальных средств.
 - c) Процесс записи информации на некотором материальном носителе.
 - d) Процесс получения логически стройных структур данных (информации) из разрозненных наборов фактов.
 - e) Отбор необходимых сведений, удовлетворяющих определенному условию.

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитать предложенные вариант-ты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).

- 2) В чем суть свойства массовости алгоритмов?
 - a) Последовательность шагов алгоритма детерминирована, т.е. после каждого шага указывается, какой шаг следует выполнять дальше, либо указывается, когда следует работу алгоритма считать законченной.
 - b) Можно применять один и тот же алгоритм для решения целого класса однотипных задач, различающихся исходными данными.
 - c) Алгоритм для размещения данных требует конечного объема памяти.
 - d) Выполнение алгоритма при любых исходных данных за конечное число шагов.

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать несколько правильных ответов.

4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

3) Назовите свойства алгоритма.

- a) Массовость
- b) Конечность
- c) Однозначность
- d) Результативность
- e) Реализуемость
- f) Бесконечность
- g) Надежность
- h) Кодированность
- i) Многозначность
- ж) Алгоритм для размещения данных требует памяти

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитать предложенные варианты-ты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).

4) Чем отличаются операторы = и == в *Python*?

- a) = (сравнение), == (присвоение)
- b) = (присвоение), == (сравнение)
- c) = (удаление), == (сравнение)
- d) = (присвоение), == (отрицание)

5) Что будет, если выполнить код ниже?

```
a = 1 == 1 or 1 != 1
b = a == 1 and a != 1

if b:
    print(b)
else:
    print(a)
```

- a) True

- b) *False*
- c) 1
- d) 0

Задание закрытого типа на установление соответствия

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидаются пары элементов.
2. Внимательно прочитайте оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.;
список 2 – утверждения, свойства объектов и т.д.
3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов.
4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).

б) Установите соответствие между оператором и выполняемым действием.

1) %	a) Тернарная операция, аналог условной конструкции if
2) ?	b) Остаток от деления
3) !=	c) Логическое И
4) and	d) Не равно

Задание закрытого типа на установление последовательности

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается последовательность элементов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Построить верную последовательность из предложенных элементов.
4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БАА или 135).

- 7) Объявите цикл перебора и вывода символов в строке *text = "Hello, World!"* в *Python*?
- a) *for*
 - b) *char*
 - c) *in*
 - d) *text:*
 - e) *print(char)*.

Тема 2 - Структуры и типы данных. Реализация базовых алгоритмов. ОПК ОС-6.2.

Контрольная работа

Вариант 1

1. Разработать блок-схему алгоритма вычисления или среднего арифметического если вводимое пользователем число a положительное или геометрического двух чисел в противном случае.
2. Нарисовать блок-схему алгоритма вычисления арифметического выражения:

$$Z = \begin{cases} 10,5x^2 + 1, & \text{если } x < 0; \\ 5(x+1)^3, & \text{если } 0 \leq x < 1. \\ \sqrt{x^2 + 1}, & \text{если } x \geq 1 \end{cases}$$

3. Нарисовать блок-схему алгоритма вычисления арифметического выражения. Значение x вводится пользователем:

$$A = \sum_{k=1}^5 \sqrt[4]{1000k + x^4}.$$

4. Поменяйте значение переменной `age`, уменьшив его на 10.

```
age = int(input())
```

```
#место для вашего кода
```

```
print(age)
```

5. Масса ископаемого диплодока Доки была M тонн, а масса нашего современника червячка Емели m грамм. Выразите в граммах массу диплодока Доки и в тоннах массу червячка Емели.

```
M = float(input())
```

```
m = float(input())
```

```
# ваш код
```

```
print(M)
```

```
print(m)
```

6. Проверка чисел. Написать программу, которая проверяет, попадает ли введенное пользователем число в диапазон от 1 до 100 включительно, и вывести "yes", если попадает. В противном случае вывести "no".

```
number = int(input("number: "))
```

```
# место для вашего кода
```

7. Проверка аргументов. Напишите функцию, которая проверяет, является ли указанный числовой аргумент четным (True) или нет (False), и

выведите соответствующее значение на экран. Числовой аргумент получите от пользователя.

```
number = int(input("number = "))  
# место для вашего кода  
print("number is even to 2:", is_even(number))
```

8. Последовательные натуральные числа. Дополните программу так, чтобы на вход она получала натуральное число n , а на выходе вычисляла и печатала сумму n последовательных натуральных чисел.

```
n = int(input()) # ввод количества слагаемых  
s = 0 # переменная для записи суммы
```

9. Список чисел. На вход программы подается число N . В следующем коде замените символы «???» так, чтобы полученный код создавал список чисел `numeric_list` от 1 до N включительно, и выведите список на экран.

```
N = int(input())  
numeric_list = ???  
for i in range(1, ???):  
    ???  
print(numeric_list)
```

10. Программа содержит две функции *GenList* и *SumList*. Функция *GenList* принимает один параметр (число) и возвращает список от 0 до значения параметра включительно. Функция *SumList* принимает в качестве параметра список и возвращает сумму четных элементов списка. Программе с помощью пользовательского ввода передается число, используемое для генерации списка функцией *GenList*. Сгенерированный список передается на вход функции *SumList*. Реализуйте пользовательский ввод, указанные функции и выведете рассчитанную сумму списка на экран.

11. В программе задан список [1, -2, 5, 8, -4, 3, -6, 7, -9, 11]. Рассчитайте сумму элементов списка, расположенных между первым и последним отрицательными элементами.

12. В программе задан список [1, -2, 5, 8, -4, 3, -6, 7, -9, 3, 2, 0, -1, 11]. Сформируйте новый список, включающий элементы исходного списка, модуль значения которых находится в интервале [2, 4].

13. В программе задан список [1, -2, 5, 8, -4, 3, -6, 7, -9, 3, 2, 0, -1, 11]. Создайте кортеж, содержащий элементы исходного списка таким образом, чтобы сначала располагались все положительные элементы, а потом - все отрицательные (элементы, равные 0, считать положительными).

14. Вам дается целое число n . Выведите матрицу размера $n \times n$, в которой все элементы 0, кроме главной диагонали, на которой стоят 1. (Главная

диагональ квадратной матрицы состоит из элементов с индексами i и j , в которых $i=j$. То есть это элементы с индексами 00, 11, 22 и т.д.).

15. Проверка аргументов. Напишите функцию `is_even`, которая проверяет, является ли указанный числовой аргумент четным (`True`) или нет (`False`), и выведите соответствующее значение на экран. Числовой аргумент получите от пользователя.

```
number = int(input("number = "))
# место для вашего кода
print("number is even to 2:", is_even(number))
```

16. В программе задан массив `[1, -2, 5, 8, -4, 3, -6, 7, -9, 3, 2, 0, -1, 11]`. Сформируйте новый массив, включающий элементы исходного массива, модуль значения которых находится в интервале `[2, 4]`. Для решения задачи используйте цикл `while`.

17. В программе задан список `[5, 3, -80, 2, 5, -80, 5999]`. Рассчитайте сумму элементов списка, расположенных между первым и последним отрицательными элементами.

18. Поиск позиции. На вход подается массив, состоящий из целочисленных элементов, упорядоченных по возрастанию. Также подается целое число x , известно, что в массиве нет элементов, равных по значению x . С помощью бинарного поиска найдите позицию (индекс), на которую нужно вставить элемент x (подвинув элементы, стоящие справа), чтобы сохранить упорядоченность массива.

```
arr = list(map(int, input().split()))
left, right = 0, len(arr)
x = int(input())
while #ваш код
    middle = left + (right - left) // 2 # середина списка
    #ваш код
```

19. Наибольшая разность. За один обход найдите максимальную разность между двумя значениями массива. Массив состоит из неповторяющихся целых чисел от 1 до 100 включительно. Разрешено использовать только один цикл для обхода массива.

```
arr = [5, 1, 3, 10, 12, 4, 15, 6]
```

20. Простая сортировка. Дан неупорядоченный массив элементов. Реализуйте простую сортировку вставкой и выведите получившийся массив в качестве ответа.

```
arr = [20, -89, -74, -72, -70, -70, -56, -14, -13, -13, -8, -2,
4, 19, 20, 28, 36, 46, 47, 58, 75, 77, 77, 81, 84, 87, 87, 97,
100]
```

Вариант 2

1. Разработать блок-схему алгоритма вычисления высоты и периметра равнобедренного треугольника, если даны его стороны.
2. Нарисовать блок-схему алгоритма вычисления арифметического выражения. Значение x вводится пользователем:

$$Z = \begin{cases} x^4 - 12, & \text{если } x \leq -1; \\ (2x+1)^3, & \text{если } -1 < x < 1; \\ \sqrt{\frac{x+11}{x}}, & \text{если } x \geq 1 \end{cases}.$$

3. Нарисовать блок-схему алгоритма вычисления арифметического выражения. Значение x вводится пользователем:

$$A = \sum_{k=1}^5 \sqrt[4]{1000k + x^4}.$$

4. Создайте переменную `value`, присвойте ей значение 36.6, а потом измените его на строку "test", введенную пользователем.

```
#место для вашего кода
```

```
print(value)
```

```
#место для вашего кода
```

```
print(value)
```

5. Найдите остаток от деления числа 1024 на 55, результат запишите в переменную `res1`. Найдите целую часть от деления числа 1024 на 55, результат запишите в переменную `res2`.

```
a = int(input())
```

```
b = int(input())
```

6. Сравнение чисел. Написать программу, которая принимает на вход от пользователя 2 числа и выводит на экран большее из них. Если числа равны — на экран должно быть выведено значение `None`.

```
one = int(input("one: "))
```

```
two = int(input("two: "))
```

```
# место для вашего кода
```

7. Проверка введенных значений. Напишите функцию, которая проверяет, являются ли введенные пользователем значения одинаковыми (True) или нет (False), и выведите соответствующее логическое значение на экран.

```
one = input("one = ")
two = input("two = ")
# место для вашего кода
print("Values are equal:", are_equal(one, two))
```

8. Учеба. На вход подается слово, выведите его буквы с соответствующими порядковыми номерами в нем (начиная с 1).

```
txt = input()
#ваш код
```

9. Пропущенные буквы в алфавите. Напишите программу, которая получает на вход список из латинских букв и добавляет в него недостающие буквы латинского алфавита. Производить сортировку полученного массива в алфавитном порядке не нужно.

```
alphabet = list("abcdefghijklmnopqrstuvwxyz")
#место для вашего кода
```

10. Программа содержит две функции *GenList* и *SumList*. Функция *GenList* принимает один параметр (число) и возвращает список от 0 до значения параметра включительно. Функция *SumList* принимает в качестве параметра список и возвращает сумму нечетных элементов списка. Программе с помощью пользовательского ввода передается число, используемое для генерации списка функцией *GenList*. Сгенерированный список передается на вход функции *SumList*. Реализуйте пользовательский ввод, указанные функции и выведите рассчитанную сумму списка на экран.
11. В программе задан список [1, -2, 5, 8, -4, 3, -6, 7, -9, 11]. Посчитайте произведение элементов списка, расположенных между максимальным по модулю и минимальным по модулю элементами.
12. В программе задан список [1, -2, 5, 8, -4, 3, -6, 7, -9, 3, 2, 0, -1, 11]. Обнулите элементы списка, модуль значения которых находится в интервале [2, 4].
13. В программе задан список [1, -2, 5, 8, -4, 3, -6, 7, -9, 3, 2, 0, -1, 11]. Создайте кортеж, содержащий элементы исходного списка таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине — элементы, стоявшие в четных позициях.
14. В этом задании вам предстоит раскрасить шахматную доску размера $n \times n$ в черный и белый цвета. Элемент с индексом [0][0] раскрасьте в белый цвет, а далее в шахматном порядке. Для обозначения черного цвета используйте

символ "b", для обозначения белого цвета — символ "w". (Подсказка - сумма $i+j$ для белых клеток четное, для черных нечетное)

15. Функция для принятия целых чисел. Напишите функцию `div`, которая принимает целое число (предварительно полученное от пользователя) и выводит на экран значения его делителей до 10. Например, для 6 будет вывод: 2, 3, 6; а для 20 будет вывод: 2, 4, 5, 10.
16. В программе задан массив `[1, -2, 5, 8, -4, 3, -6, 7, -9, 11]`. Рассчитайте сумму элементов массива, модуль которых больше 4. Для решения задачи используйте цикл `while`.

Формат вывода: *"делители числа: n"*, где n - соответствующий делитель. Выводите каждый делитель в отдельной строке.

```
number = int(input("number = "))  
# место для вашего кода
```

17. В программе задан список `[5, 3, -80, 2, 5, -80, 99]`. Сформируйте новый список, включающий элементы исходного списка, модуль значения которых находится в интервале `[2, 81]`.
18. Единственное пропущенное число. Дан упорядоченный целочисленный массив длины N , содержащий уникальные значения в диапазоне от 0 до N включительно. С помощью бинарного поиска найдите единственное пропущенное число из диапазона 0- N .

```
arr = list(map(int, input().split()))  
left, right = 0, len(arr)  
while #ваш код:
```

```
    middle = left + (right - left) // 2
```

19. Линейный поиск. На вход подается массив, состоящий из целочисленных элементов, упорядоченных по возрастанию. Также подается целое число x , известно, что в массиве нет элементов, равных по значению x . С помощью линейного поиска найдите позицию (индекс), на которую нужно вставить элемент x (подвинув элементы, стоящие справа), чтобы сохранить упорядоченность массива.

```
arr = [1, 40, 75, 89, 302, 302, 699, 701, 899, 1000, 700]
```

20. Простая сортировка. Дан неупорядоченный массив элементов. Реализуйте простую сортировку выбором и выведите получившийся массив в качестве ответа.

```
arr = [20, -89, -74, -72, -70, -70, -56, -14, -13, -13, -8, -2,  
4, 19, 20, 28, 36, 46, 47, 58, 75, 77, 77, 81, 84, 87, 87, 97,  
100]
```

Тема 3 - Основы *Data Science*. Использование языка программирования для решения экономических задач. ОПК ОС-6.2.

Практическое контрольное задание

Задание 1. Анализ временных рядов

Вариант 1

Динамика потребления овощей на одного члена домохозяйства в области за 1993-2001 гг. характеризуется следующими данными:

Годы	2017	2018	2018	2019	2020	2021	2022	2023	2024
Потребление овощей, кг.	30,0	32,1	36,0	30,9	38,7	48,9	46,8	53,4	54,0

Произведите сглаживание временного ряда Y_t методом скользящих средних, используя простую среднюю арифметическую с интервалом сглаживания $m=3$. Проверьте данные на аномальность.

Определите прогноз на следующий год.

Вариант 2

Имеются данные об урожайности озимой пшеницы y_t за 10 лет:

t	1	2	3	4	5	6	7	8	9	10
y_t	16,2	20,1	17,2	7,8	15,2	16,2	19,7	14,5	18,6	20,4

Произведите сглаживание временного ряда y_t методом скользящих средних, используя простую среднюю арифметическую с интервалом сглаживания $m=3$. Проверьте данные на аномальность.

Определите прогноз на следующий год.

Задание 2. Визуализация данных

Вариант 1

- 1) Постройте графики функций $y=x^2$ и $y=tg(x)$ на одной координатной прямой
- 1) Постройте столбчатую диаграмму по следующим данным $companies = ['Газпром', 'ВТБ', 'Сбер', 'Роснефть', 'Лукойл']$ $counts = [42, 34, 41, 31, 20]$
- 2) Постройте круговую диаграмму по следующим данным $vals = [25, 17, 36, 15, 21]$ $labels = ['Microsoft', 'Aquarius', 'Astra', 'TRU', 'Dell']$.
- 2)

Вариант 2

- 3) Постройте графики функций $y=-x^2$ и $y=ctg(x)$ на одной координатной прямой

- 4) Постройте столбчатую диаграмму по следующим данным *companies* = ['Спб', 'Москва', 'Мурманская область', 'Башкортостан', 'Ленинградская область'] *counts* = [5; 15; 0,3; 3,1; 2]
- 5) Постройте круговую диаграмму по следующим данным *vals* = [24, 17, 53, 21, 35] *labels* = ['Ford', 'Toyota', 'BMW', 'AUDI', 'Jaguar'].

Тема 4- Графовые алгоритмы и анализ экономических сетей. ОПК ОС-6.2.

Тест 4.1

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитайте предложенные вариант-ты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).

- 1) Что позволяет определить алгоритм Форда-Фалкерсона?
 - a) Кратчайший путь между вершинами
 - b) Максимальный поток сети
 - c) Решение задачи коммивояжера

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитайте предложенные вариант-ты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).

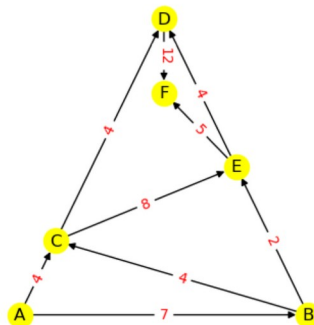
- 2) Что такое суммарный чистый поток?

- a) разность суммарного положительного потока, выходящего из данной вершины, и суммарного положительного потока, входящего в нее.
- b) свойство определяющее, что поток в транзитной вершине должен быть равен 0
- c) сумма потоков, выходящих из данной вершины

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа.
5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).

3) Вычислите максимальный поток в сети из A в F .



- a) 4
- b) 10
- c) 20
- d) 5

Задание закрытого типа на установление соответствия

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов.
2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.;
список 2 – утверждения, свойства объектов и т.д.
3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов.

4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).

4) Установите соответствие между алгоритмами и их действиями

1) Алгоритм Дейкстры	а) Поиск кратчайшего пути во взвешенном графе с положительными весами
2) Алгоритм Крускала	б) Поиск минимального остовного дерева
3) Алгоритм Флойда	с) Поиск кратчайшего пути во взвешенном графе с положительными и отрицательными весами
4) Алгоритм Форда-Фалкерсона	д) Алгоритм поиска максимального потока

Задание закрытого типа на установление последовательности.

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается последовательность элементов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Построить верную последовательность из предложенных элементов.
4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).

5) Добавьте вершину в граф с атрибутом *Time*.

- a) G.
- b) add_node
- c) (1,
- d) Time
- e) ='5pm'
- f))

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных

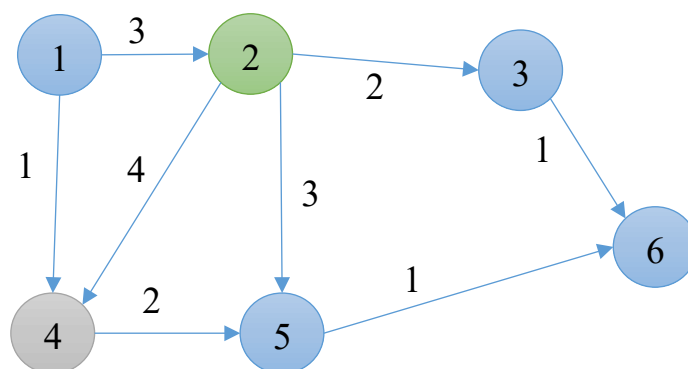
1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

- б) Перечислите ограничения алгоритма Дейкстры:
- Граф должен быть взвешенным.
 - Граф не должен быть взвешенным
 - Весы должны быть неотрицательными.
 - Весы должны быть отрицательными.

Задание открытого типа с развернутым ответом.

- Внимательно прочитайте текст задания и понять суть вопроса.
- Продумать логику и полноту ответа.
- Записать ответ, используя четкие компактные формулировки.
- В случае расчетной задачи, записать решение и ответ/

Построить взвешенный граф, изображенный на рисунке. Рассчитать и вывести на экран степень каждой вершины для неориентированных графов и степень каждой вершины по выходным ребрам для орграфов. Вывести на экран список смежных ребер каждой вершины графа. Рассчитать кратчайший путь из вершины 1 в вершину 6. Вычислить максимальный поток из 1 в 6.



5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных точек):
приведены в п.6.2.

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной точкой (далее – КТ). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КТ в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КТ составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КТ в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной точки	Максимальное количество баллов за работу в рамках КТ, которое может набрать студент	Коэффициент веса контрольной точки	Результат контрольной точки, участвующий в формировании итоговой балльной оценки по дисциплине (отражается в журнале БРС в СДО)
1 семестр			
КТ - 1 Тест 1.1 - Информационные процессы, основы алгоритмизации	100	0,3	30
КТ – 2 Контрольная работа - Структуры и типы данных. Реализация базовых алгоритмов.	100	0,3	30
Итого:	x	0,6	60
2 семестр			
КТ - 3 ПКЗ 3.1 - Основы <i>Data Science</i> . Использование языка программирования для решения экономических задач.	100	0,3	30
КТ - 4 Тест 4.1 - Графовые алгоритмы и анализ экономических сетей	100	0,3	30
Итого:	x	0,6	60

Формула расчета результата контрольной точки:

Результат контрольной точки = Количество баллов за работу в рамках КТ x Коэффициент веса контрольной точки.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КТ и типовые оценочные материалы:

1 семестр

КТ-1: Тема 1. Тестирование

КТ-2: Тема 2. Контрольная работа.

2 семестр

КТ-3: Тема 3. Практическое контрольное задание (ПКЗ)

КТ-4: Тема 4. Тестирование

Для каждой формы текущего контроля успеваемости обучающихся в рамках КТ определены критерии оценивания результатов выполнения задания.

1. Критерии оценивания тестирования:

Критерии оценки	Диапазон	Описание критерия
-----------------	----------	-------------------

	баллов	
<i>Количество правильных ответов</i>	0	<i>Количество правильных ответов менее 55%</i>
	25	<i>Количество правильных ответов от 55% до 64%</i>
	50	<i>Количество правильных ответов от 65% до 74%</i>
	75	<i>Количество правильных ответов от 75% до 84%</i>
	100	<i>Количество правильных ответов от 85% до 100%</i>
Итого максимально:	100	

2. Критерии оценивания ПКЗ:

Критерии оценки	Диапазон баллов	Описание критерия
<i>Содержание и раскрытие выбранных понятий</i>	41-70	<i>Детальное описание всех понятий на примере разработанной программы</i>
	21-40	<i>Поверхностное описание всех понятий на примере разработанной программы</i>
	0-20	<i>Понятия раскрыты минимально или не раскрыты вовсе</i>
<i>Количество выполненных заданий</i>	30	<i>Количество выполненных заданий от 85% до 100%</i>
	15	<i>Количество выполненных заданий от 55% до 84%</i>
	0	<i>Количество выполненных заданий менее 55%</i>
Итого максимально:	100	

3. Критерии оценивания КР

Критерии оценки	Диапазон баллов	Описание критерия
<i>Содержание и раскрытие выбранных понятий</i>	41-70	<i>Детальное описание всех понятий на примере разработанной программы</i>
	21-40	<i>Поверхностное описание всех понятий на примере разработанной программы</i>
	0-20	<i>Понятия раскрыты минимально или не раскрыты вовсе</i>
<i>Количество выполненных</i>	30	<i>Количество выполненных заданий от 85% до 100%</i>

заданий	15	Количество выполненных заданий от 55% до 84%
	0	Количество выполненных заданий менее 55%
Итого максимально:	100	

5.5. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий (*при необходимости*).

Для решения задач открытого типа (кейсов, ПКЗ), тестовых заданий студенту разрешается использование онлайн компиляторов, справочной системой *Python* и других сторонних сайтов. Для построения блок-схем алгоритмов студенту можно использовать любой соответствующий онлайн-инструмент.

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация проводится в форме:

1 семестр - экзамен; 2 семестр - экзамен.

Экзамен проводится в компьютерном классе в письменной форме. Обучающийся получает экзаменационный билет с вариантами 3-х заданий различного типа. На выполнение заданий дается 40-60 минут. По завершении подготовки необходимо представить ответы в письменном виде, подробно изложив ход выполнения задания, сделать выводы (*при необходимости*).

При реализации промежуточной аттестации в ЭО/ДОТ могут быть использованы следующие формы: устно в ДОТ - в форме обоснованных ответов на задания различного типа; письменно в СДО - в форме письменного решения заданий различного типа; тестирование в СДО.

6.2. Типовые оценочные материалы промежуточной аттестации.

Вопросы для подготовки к экзамену 1 семестра.

1. Информационный процесс. Понятие алгоритма. Свойства алгоритмов.
2. Применение алгоритмов. Связь между алгоритмом и понятием функции.
3. Итерация, рекуррентность и рекурсивность. Инвариант в программировании, верификация программ.
4. Документирование, тестирование и верификация программного кода.
5. Технологии разработки программ. Гибкие технологии разработки. Экстремальное программирование.
6. Технические и программные средства реализации информационных процессов.

7. Сложность по времени и используемой памяти. Оптимальность. Трудоемкость алгоритмов.
8. Типы данных и операции с типами данных в языке программирования. Операторы сравнения. Сложные логические операции. Операции со строками, индексы и срезы.
9. Определение и основные характеристики списков. Индексы и срезы.
10. Определение и основные характеристики кортежа, операции с кортежами.
11. Множества, основные операции с множествами.
12. Основные сведения о словарях, основные операции со словарями.
13. Структуры данных для ассоциированных списков. Простая реализация ассоциативных массивов.
14. Хеш-функции, хеш-суммы, хеш-таблица. Коллизии. Алгоритм открытой адресации.
15. Линейный поиск и бинарный поиск.
16. Основы ООП.
17. Абстрактные структуры данных: стек, очередь, очередь с приоритетом, ассоциативный массив.
18. Пузырьковая сортировка. Сортировка выбором и сортировка слиянием.
19. Быстрая сортировка и пирамидальная сортировка.
20. *Tim Sort*.

Типовые задания для экзамена в первом семестре.

1. Напишите программу для подсчета среднего арифметического значения всех введенных пользователем чисел $Ср.ариф. = \frac{x_1 + x_2 + \dots + x_n}{n}$. Индикатором окончания ввода будет служить ноль. При этом программа должна выдавать соответствующее сообщение об ошибке, если первым же введенным пользователем значением будет ноль.
2. Напишите функцию, принимающую на вход длины двух катетов прямоугольного треугольника и возвращающую длину гипотенузы, рассчитанную по теореме Пифагора. В главной программе должен осуществляться запрос длин сторон у пользователя, вызов функции и вывод на экран полученного результата.
3. Напишите программу, которая будет запрашивать у пользователя целочисленные значения и сохранять их в виде списка. Индикатором окончания ввода значений должен служить ноль. Затем программа должна вывести на экран все введенные пользователем числа (кроме нуля) в порядке возрастания – по одному значению в строке. Используйте для сортировки либо метод *sort*, либо функцию *sorted*.
4. Напишите программу, отображающую последовательность кнопок, которую необходимо нажать, чтобы на экране телефона появился текст, введенный пользователем. Создайте словарь, сопоставляющий символы с

кнопками, которые необходимо нажать, а затем воспользуйтесь им для вывода на экран последовательности кнопок в соответствии с введенным пользователем сообщением по запросу. Например, на ввод строки *Hello, World!* ваша программа должна откликнуться следующим выводом: 443355 5555666110966677755531111.

5. Сумма первых n положительных чисел. Напишите программу, запрашивающую у пользователя число и подсчитывающую сумму натуральных положительных чисел от 1 до введенного пользователем значения. Сумма первых n положительных чисел может быть рассчитана по формуле: $\sum_{i=1}^n \frac{n(n+1)}{2}$.

1. Дан словарь с записями, ключами которых являются числовые id , а значениями – города.

```
cities = {100: 'Москва', 101: 'Санкт-Петербург', 103: 'Екатеринбург', 104: 'Пермь',  
107: 'Красноярск', 111 : 'Киров'}
```

Напишите программу, которая просит пользователя ввести с клавиатуры сначала ключ, потом значение, а затем: если в словаре *cities* есть запись с таким ключом, на экран выводится сообщение «Запись с таким ключом существует. Введите другой ключ.» Если в словаре *cities* нет записи с таким ключом, в этот словарь добавляется запись с введенным ключом и значением и обновленный словарь *cities* выводится на экран.

Вопросы для подготовки к экзамену 2 семестра.

1. Связность, ориентированные графы. Поиск в ширину.
2. Волновой алгоритм.
3. Алгоритм Прима.
4. Алгоритм Крускала.
5. Сравнение алгоритмов (Волновой, Прима, Крускала), особенности реализации и область применимости, оценка их трудоемкости.
6. Алгоритм Флойда.
7. Алгоритм Литла.
8. Алгоритм Дейкстры.
9. Сравнение алгоритмов (Флойда, Литла, Дейкстры), особенности реализации и область применимости, оценка их трудоемкости.
10. Задача Коммивояжера.
11. Применение графов для решения транспортной задачи.
12. Алгоритм Форда-Фалкерсона и максимальный поток.
13. Основные задачи раскраски графов. Методы решения и сфера применения.
14. Критический путь в графе.
- 15.. Создание массивов. Атрибуты массивов *NumPy*.

16. Создание коллекций. Обращение к элементам коллекций. Описательная статистика коллекции.
17. Фреймы данных. Создание фрейма данных, обращение к столбцам и строкам фрейма данных.
18. Доступ к файловым данным. Чтение и запись текстовых данных.
19. Визуализация данных. Линейные диаграммы. Визуализация погрешностей. Столбчатые диаграммы. Круговые диаграммы. Гистограммы.
20. Регулярные и нерегулярные временные ряды. Общие методы анализа временных рядов.
21. Введение в машинное обучение. Алгоритм кластеризации k-средних.

Типовые задания для экзамена во втором семестре.

1. Постройте на одной координатной плоскости графики функций $y=|x|$ и $y=x^{\frac{1}{2}}$
2. В таблице представлены квартальные данные об объеме производства в денежном выражении (ден.ед.)

Квартал	1-й квартал 2006	2-й квартал 2006	3-й квартал 2006	4-й квартал 2006	1-й квартал 2007	2-й квартал 2007
Y_t	21,18	22,23	23,26	24,31	25,34	26,37

Определите прогноз производства в 3-м квартале 2007 года с помощью среднего прироста.

3. Создайте *DataFrame* из словаря списков.

```
import pandas as pd
data = {'X': [1, 2, 3, 4], 'Y': [5, 6, 7, 8]}
```

Решение

```
df = pd.DataFrame(data)
print(df)
```

4. Выберите первые 3 строки фрейма данных

```
import pandas as pd
data = {'X': [1, 2, 3, 4], 'Y': [5, 6, 7, 8]}
df = pd.DataFrame(data)
print(df.head(3))
```

Решение

```
print(df.head(3))
```

5. Фильтруйте строки на основе условия $x > 2$.

```
import pandas as pd
data = {'X': [1, 2, 3, 4], 'Y': [5, 6, 7, 8]}
df = pd.DataFrame(data)
```

Решение

```
filtered_df = df[df['X'] > 2]
print(filtered_df)
```


6. Добавьте новый столбец в существующий фрейм данных.

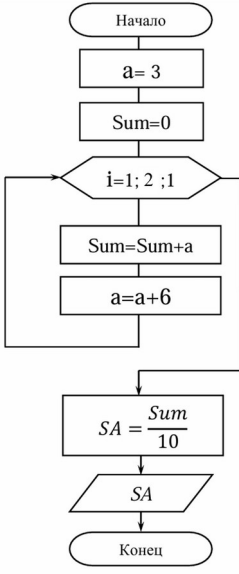
```
import pandas as pd
data = {'X': [1, 2, 3, 4], 'Y': [5, 6, 7, 8]}
df = pd.DataFrame(data)
```

Решение

```
df['Z'] = df['X'] + df['Y']
print(df)
```

Типовые проверочные задания для самоподготовки обучающегося к промежуточной аттестации в 1 семестре:

ТИП ЗАДАНИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	ТИПОВЫЕ ЗАДАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, А или В).	1. Какое логическое выражение истинно, если $x \in [-10, 10]$? а) $(x > 10) \text{ И } (x < -10)$ б) $(x > 10) \text{ ИЛИ } (x < -10)$ в) $(x < 10) \text{ ИЛИ } (x > -10)$ г) $(x > -10) \text{ И } (x < 10)$
		2. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?  а) разветвляющийся с неполным ветвлением б) разветвляющийся с полным ветвлением

		<p>в) цикл с заданным условием окончания работы г) цикл с заданным числом повторений д) цикл с предусловием</p> <p>3. Исполните фрагмент алгоритма при $a = 2$ и $b = 0$.</p>  <pre> graph TD Start([Начало]) --> A[a=3] A --> Sum[Sum=0] Sum --> Loop{i=1; 2; 1} Loop --> SumAdd[Sum=Sum+a] SumAdd --> AAdd[a=a+6] AAdd --> Loop Loop --> SA[SA = Sum/10] SA --> SAOut[/SA/] SAOut --> End([Конец]) </pre> <p>Определите значение переменной SA после выполнения фрагмента алгоритма.</p> <p>а) 0,6 б) 1,2 в) 1,8 г) 1,9 д) 2,0 е) 0,7</p>																						
<p>Задание закрытого типа на установление соответствия</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) варианты ответа (например, А1 или Б4).</p>	<p>1. Установите соответствие между оператором и выполняемым действием.</p> <table border="1" data-bbox="853 1198 1444 1388"> <tr> <td>1) %</td> <td>а) Тернарная операция, аналог условной конструкции <i>if</i></td> </tr> <tr> <td>2) ?</td> <td>б) Остаток от деления</td> </tr> <tr> <td>3) !=</td> <td>с) Логическое И</td> </tr> <tr> <td>4) &&</td> <td>д) Не равно</td> </tr> </table> <p>2. Установите соответствие между оператором и выполняемым действием.</p> <table border="1" data-bbox="853 1444 1444 1753"> <thead> <tr> <th>Операция</th> <th>Приоритет</th> </tr> </thead> <tbody> <tr> <td>1. $-i$</td> <td>а) инкремент</td> </tr> <tr> <td>2.</td> <td>б) выделение памяти</td> </tr> <tr> <td>3. new</td> <td>с) инверсия</td> </tr> <tr> <td>4. $i=i$</td> <td>д) умножение</td> </tr> <tr> <td>5. i</td> <td>е) проверка на эквивалентность</td> </tr> <tr> <td>6. $++i$</td> <td>ф) вычитание с присвоением</td> </tr> </tbody> </table>	1) %	а) Тернарная операция, аналог условной конструкции <i>if</i>	2) ?	б) Остаток от деления	3) !=	с) Логическое И	4) &&	д) Не равно	Операция	Приоритет	1. $-i$	а) инкремент	2.	б) выделение памяти	3. new	с) инверсия	4. $i=i$	д) умножение	5. i	е) проверка на эквивалентность	6. $++i$	ф) вычитание с присвоением
1) %	а) Тернарная операция, аналог условной конструкции <i>if</i>																							
2) ?	б) Остаток от деления																							
3) !=	с) Логическое И																							
4) &&	д) Не равно																							
Операция	Приоритет																							
1. $-i$	а) инкремент																							
2.	б) выделение памяти																							
3. new	с) инверсия																							
4. $i=i$	д) умножение																							
5. i	е) проверка на эквивалентность																							
6. $++i$	ф) вычитание с присвоением																							
<p>Задание закрытого типа с выбором нескольких правильных ответов из нескольких предложенных</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько правильных ответов.</p>	<p>1. Выберите верные утверждения про атрибуты экземпляра класса и атрибуты класса <i>Python</i>.</p> <p>а) <u>Атрибуты класса — это данные, которые описывают класс. Эти данные присущи самому классу и по умолчанию будут отображаться для всех созданных экземпляров</u> б) Они ничем не отличаются друг от друга, это одно и то же с) <u>Атрибуты экземпляра класса — это</u></p>																						

	<p>4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).</p>	<p><u>данные, присущие конкретному экземпляру класса или объекту</u></p> <p>d) Атрибуты класса — это данные, присущие конкретному экземпляру класса</p> <p>е) д. Атрибуты экземпляра класса — это данные, которые описывают класс. Эти данные присущи самому классу и по умолчанию будут отображаться для всех созданных экземпляров</p> <p>2. Что такое метод в ООП и чем он отличается от функции?</p> <p>а) Ничем не отличается</p> <p>б) <u>Метод — это функция, объявленная в теле класса</u></p> <p>с) <u>Метод привязан к конкретному объекту и определяет его поведение</u></p> <p>д) Метод не привязан к конкретному объекту</p>
<p>Задание закрытого типа на установление последовательности</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Построить верную последовательность из предложенных элементов.</p> <p>4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).</p>	<p>1. Определите функцию принимающую один параметр и возвращающую квадрат этого параметра.</p> <p>Ключевые слова:</p> <p>a) <i>def</i></p> <p>b) <i>function</i></p> <p>c) <i>(number)</i></p> <p>d) :</p> <p>e) <i>return</i></p> <p>f) <i>number</i></p> <p>g) <i>*number</i></p>
<p>Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Выбрать один верный ответ.</p> <p>4. Записать только номер (или букву) выбранного варианта ответа.</p> <p>5. Записать аргументы, обосновывающие выбор ответа.</p>	<p>1. Какая сложность у алгоритма, реализуемого кодом ниже</p> <pre style="background-color: #f0f0f0; padding: 10px;"> a = [5, 9, 7, 3, 6, 4] to_find = 4 found = False for el in a: if el == to_find: found = True break </pre> <p>a) $O(1)$</p> <p>b) $O(n)$</p> <p>c) $O(n^2)$</p> <p>d) $O(\log n)$.</p> <p>2. Какая сложность у алгоритма, реализуемого кодом ниже</p> <pre style="background-color: #f0f0f0; padding: 10px;"> a = [5, 9, 7, 3, 6, 4] for i in range(len(a) - 1): for j in range(len(a) - i - 1): if a[j] > a[j + 1]: a[j], a[j + 1] = a[j + 1], a[j] </pre>

		<pre>print(a)</pre> <p>а. $O(1)$ б. $O(n)$ в. $O(n^2)$ г. $O(\log n)$ а)</p>
Задание открытого типа с развернутым ответом	<ol style="list-style-type: none"> Внимательно прочитать текст задания и понять суть вопроса. Продумать логику и полноту ответа. Записать ответ, используя четкие компактные формулировки. В случае расчетной задачи, записать решение и ответ. 	<ol style="list-style-type: none"> Напишите программу, которая будет запрашивать у пользователя целочисленные значения и сохранять их в виде списка. Индикатором окончания ввода значений должен служить ноль. Затем программа должна вывести на экран все введенные пользователем числа (кроме нуля) в порядке возрастания – по одному значению в строке. Используйте для сортировки либо метод <i>sort</i>, либо функцию <i>sorted</i>. Напишите функцию с именем <i>precedence</i>, которая будет возвращать целое число, представляющее собой приоритет или старшинство математического оператора. В качестве единственного параметра эта функция будет принимать строку, содержащую оператор. На выходе функция должна давать 1 для операторов «+» и «-», 2 для «*» и «/» и 3 для «^». Если строка, переданная в функцию, не содержит ни один из перечисленных операторов, должно быть возвращено значение -1.

Типовые проверочные задания для самоподготовки обучающегося к промежуточной аттестации во 2 семестре:

ТИП ЗАДАНИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	ТИПОВЫЕ ЗАДАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких предложенных	<ol style="list-style-type: none"> Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. Внимательно прочитать предложенные варианты ответа. Выбрать один верный ответ. Записать только номер (или букву) выбранного варианта ответа (например, А или В). 	<ol style="list-style-type: none"> Найти максимальный поток (1 → 4) с помощью алгоритма Форда-Фалкерсона <div style="text-align: center;"> <pre> graph LR 1((1)) -- 7 --> 2((2)) 1((1)) -- 4 --> 6((6)) 2((2)) -- 2 --> 3((3)) 2((2)) -- 4 --> 6((6)) 3((3)) -- 5 --> 4((4)) 3((3)) -- 4 --> 5((5)) 6((6)) -- 4 --> 5((5)) </pre> </div> <p>а) 10 б) 12 в) 34 г) 15</p> Как можно выполнить поэлементное умножение двух массивов <i>NumPy</i> <i>a</i> и <i>b</i>? <p>а) $a * b$ б) <i>np.multiply(a, b)</i> в) <i>np.mult(a, b)</i> г) <i>a.dot(b)</i></p>
Задание закрытого типа с выбором	<ol style="list-style-type: none"> Внимательно прочитать текст задания и понять, что в качестве 	<ol style="list-style-type: none"> Что делает алгоритм Беллмана-Форда в конце своего выполнения?

<p>нескольких правильных ответов из нескольких вариантов предложенных</p>	<p>ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).</p>	<p>a) Возвращает кратчайшие пути только в том случае, если существуют отрицательные циклы b) <u>Всегда возвращает кратчайший путь</u> c) <u>Выявляет негативные циклы</u></p>
<p>Задание закрытого типа на установление последовательности</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).</p>	<p>1. Удалите столбец <i>z</i> из фрейма <i>df</i>. Ключевые слова: h) <i>df</i>. i) <i>drop</i> j) (k) <i>columns</i>= l) [<i>'Z'</i>], m) <i>Inplace</i> = n) <i>True</i> o))</p>
<p>Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа.</p>	<p>1. Вычислите среднее значение значений в каждой строке.</p> <pre>import pandas as pd data = {'X': [4, 3, 2, 1], 'Y': [8, 7, 6, 5]}</pre> <p>Ответ</p> <p>a) <i>df.sort_values(by='X', inplace=True)</i> <i>print(df)</i> b) <i>print(df.mean(axis=1))</i> c) <i>pivot_table = df.pivot_table(values='Z', index='X', columns='Y')</i> <i>print(pivot_table)</i></p>
<p>Задание открытого типа с развернутым ответом</p>	<p>1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ.</p>	<p>1. Произведите сортировку фрейма данных по столбцу.</p> <pre>import pandas as pd data = {'X': [4, 3, 2, 1], 'Y': [8, 7, 6, 5]}</pre> <p>Ответ</p> <p><i>df.sort_values(by='X', inplace=True)</i> <i>print(df)</i></p> <p>2. Создайте DataFrame с определенными именами столбцов. Ответ</p>

		<pre>import pandas as pd data = {'col1': [1, 2, 3], 'col2': [4, 5, 6]} df = pd.DataFrame(data) print(df)</pre>
--	--	---

6.3. Критерии и шкала оценивания на основе БРС.

Критерии и балльная шкала определяются преподавателем

Экзамен

КРИТЕРИИ ОЦЕНИВАНИЯ	РЕЗУЛЬТАТ В БАЛЛАХ
<i>Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где он продемонстрировал знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок</i>	40
<i>Дан развернутый ответ на поставленный вопрос, где студент демонстрирует знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры, в ответе присутствует свободное владение монологической речью, логичность и последовательность ответа. Однако допускается неточность в ответе. Решил предложенные практические задания с небольшими неточностями.</i>	30-39
<i>Дан ответ, свидетельствующий в основном о знании процессов изучаемой дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры, недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа и решении практических заданий.</i>	20-29
<i>Дан ответ, который содержит ряд серьезных неточностей,</i>	0-19

<i>обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е. студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.</i>	
--	--

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий (*при необходимости*).

Для решения задач открытого типа (кейсов, ПКЗ), тестовых заданий студенту разрешается использование онлайн компиляторов и справок по языку C#. Для построения блок-схем алгоритмов студенту можно использовать любой соответствующий онлайн-инструмент.

7. Методические материалы по освоению дисциплины

Для изучения основных вопросов образовательной программы необходимо конспектировать материалы лекций, работать с рекомендованной преподавателем литературой, а также ресурсами информационно-телекоммуникационной сети «Интернет». Для приобретения навыков активного использования знаний полезно обсуждать плановые и возникающие вопросы, а также решаемые задачи на практических занятиях. Чтобы легче и прочнее усвоить материал следует постоянно использовать конкретные примеры, сравнения из уже полученных областей наук.

Для закрепления изученного материала даны вопросы по каждой теме дисциплины, на которые следует самостоятельно найти ответы.

Важной составной частью учебного процесса в вузе являются практические занятия. Практические занятия проводятся главным образом по дисциплинам, требующим закрепления навыков решения задач. Все практические занятия проводятся в компьютерных классах с использованием ИСР *JupiterNotebook* и ИСР среды *VS Studio Code*. Каждая работа должна быть защищена, т.е. студент должен ответить на вопросы преподавателя о ходе выполнения работы, а также на вопросы теоретического характера.

При подготовке к практическим занятиям необходимо проанализировать конспект лекции, ознакомиться с рекомендованной литературой по соответствующей теме, осуществить подготовку по рекомендованным в рабочей программе вопросам для обсуждения темы, выполнить домашнее задание (при необходимости).

Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе

самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале.

В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю (в том числе по электронной почте).

Планируя консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам. Записи имеют первостепенное значение для самостоятельной работы студентов. Они помогают понять построение изучаемого материала, выделить основные положения, проследить их логику. Кроме того, ведение записей способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у студента, систематически ведущего записи, создается свой индивидуальный фонд методических материалов для быстрого повторения изученных вопросов, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе.

После изучения базовых тем курса проводится текущий контроль знаний студентов в виде опроса или письменного тестирования. Типовые тесты и задания по темам дисциплины приведены в специальном разделе данной рабочей программы.

Подготовка к текущему и промежуточному контролю предполагает изучение представленных вопросов к зачету, работу над тестами, представленными в данной рабочей программе, выполнение семестровой проектной работы по применению системного подхода и методов системного анализа к выбранной системе.

Работа в малых группах – это одна из самых популярных форм проведения занятий, так как она дает всем обучающимся (в том числе и стеснительным) возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения (в частности, умение активно слушать, вырабатывать общее мнение, разрешать возникающие разногласия). Цель данной формы проведения занятий: продемонстрировать сходство или различия определенных явлений, выработать стратегию или разработать план, выяснить отношение различных групп участников к одному и тому же вопросу. В ходе этой работы дополнительно решаются следующие задачи: развитие навыков общения и взаимодействия в группе, формирование ценностно-ориентационного единства

группы, поощрение к гибкой смене социальных ролей в зависимости от ситуации.

Группа студентов делится на несколько малых групп. Количество групп определяется числом творческих заданий, которые будут обсуждаться в процессе занятия. Малые группы формируются либо по желанию студентов, либо по родственной тематике для обсуждения. Каждая малая группа обсуждает творческое задание в течение отведенного времени. Основной этап – проведение обсуждения творческого задания. Заслушиваются суждения, предлагаемые каждой малой группой по творческому заданию. Преподаватель дает оценочное суждение и работе малых групп, по решению творческих заданий, и эффективности предложенных путей решения.

С целью активизации самостоятельной работы студентов в системе дистанционного обучения *Moodle* разработан учебный курс «Программирование», включающий набор файлов с текстами лекций, заданиями для выполнения практических и лабораторных работ. Для активизации работы студентов во время контактной работы с преподавателем отдельные занятия проводятся в интерактивной форме. В основном, интерактивная форма занятий обеспечивается при проведении занятий в компьютерном классе. Интерактивная форма обеспечивается наличием разработанных файлов с заданиями, наличием контрольных вопросов, возможностью доступа к системе дистанционного обучения, а также к тестеру.

В качестве самостоятельной работы студентами выполняется семестровая работа по всем темам. При выполнении заданий по темам могут использоваться представленные студентом материалы по предыдущим темам. Выполненная семестровая работа представляется студентом на открытой защите на промежуточной аттестации.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Хайнеман Джордж: Алгоритмы. С примерами на Python. - СПб.: Питер, 2023. - 304 с.: ил. - (Серия «Бестселлеры O'Reilly»). ISBN 978-5-4461-1963-9
2. Васильев, Ю. Python для *data science*. — СПб.: Питер, 2023. — 272 с.: ил. — (Серия «Библиотека программиста»)
3. Губернаторов, А. М. Экономика на Python [Электронный ресурс] : учеб. пособие /А. М. Губернаторов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2023. – 290 с. – ISBN 978-5-9984-1873-0.
4. Бхаргава Адитья: Грокаем алгоритмы. 2-е изд. — СПб.: Питер, 2025. — 352 с.: ил. — (Серия «Библиотека программиста»).ISBN 978-5-4461-4172-2

8.2. Дополнительная литература.

1. Уэс Маккинли: *Python* и анализ данных / ДМК Пресс, 2020. – 482 с.:
2. Ахмад Имран: 40 алгоритмов, которые должен знать каждый программист на Python. — СПб.: Питер, 2023. — 368 с.
3. Костюкова, Н. И. Графы и их применение : учебное пособие / Н. И. Костюкова. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 147 с. — ISBN 978-5-4497-0367-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/89435.html>
4. Дейтел Пол, Дейтел Харви: *Python: Искусственный интеллект, большие данные и облачные вычисления*. — СПб.: Питер, 2020. — 864 с.: ил. — (Серия «Для профессионалов»)

8.3.Нормативные правовые документы и иная правовая информация

Не используются.

8.4 Интернет-ресурсы

Обучающимся обеспечен доступ к материалам курса в СДО Академии <http://lms.ranepa.ru>, а также через сайт научной библиотеки к следующим подписным электронным ресурсам:

Русскоязычные ресурсы

- Электронные учебники электронно-библиотечной системы (ЭБС) «Айбукс»
- Электронные учебники электронно-библиотечной системы (ЭБС) «Юрайт»

- Электронные учебники электронно-библиотечной системы (ЭБС) «Лань»
- Электронные учебники электронно-библиотечной системы (ЭБС) «ZNANIUM.COM»
- Электронные учебники электронно-библиотечной системы (ЭБС) «BOOK.RU»
- Электронные учебники электронно-библиотечной системы (ЭБС) «IPRSMART»

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

№ п/п	Наименование
1.	Специализированные залы для проведения лекций, оснащенные персональным компьютером/ноутбуком и мультимедийным проектором
2.	Аудитории и компьютерные классы, оборудованные посадочными местами и персональными компьютерами с выходом в Интернет для проведения практических занятий. Программное обеспечение: <i>Jupiter Notebook, MSVisualStudioCode</i>
3.	«МТС Линк» — российская платформа для онлайн-коммуникаций и совместной работы команд ; «Яндекс Телемост» — сервис для видеоконференций от Яндекса; Яндекс-мессенджер
4.	Технические средства обучения: персональные компьютеры; программные средства, обеспечивающие просмотр видеофайлов в форматах <i>AVI, MPEG-4, DivX, RMVB, WMV</i> ; программы для работы с электронными таблицами для обработки, анализа и визуализации данных; соответствующие онлайн-инструменты для построения интеллект-карты и моделей в различных нотациях
5.	Научная библиотека (в т.ч. электронные информационные ресурсы научной библиотеки)
6.	СДО Академии https://lms.ranepa.ru/