

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.06 Алгоритмизация и программирование
(индекс, наименование дисциплины в соответствии с учебным планом)

38.03.01 Экономика
(код, наименование направления подготовки)

Финансы и кредит
(наименование образовательной программы)

очная, очно-заочная формы обучения
(форма обучения)

Год набора – 2026

Киров 2026

Автор(ы)-составитель(и) РПД:

Мякишев Сергей Леонидович, к.э.н., доцент кафедры «Экономики и государственного и муниципального управления»

И.о. заведующего кафедрой:

И.О. «Правовых и общественных дисциплин» к.э.н., Русакова Елена Андреевна

Рабочая программа дисциплины Б1.О.06 Алгоритмизация и программирование одобрена на заседании кафедры «Экономики и государственного и муниципального управления» Кировского филиал РАНХиГС.

Протокол №7 от «17» декабря 2025 г..

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы	4
2. Объем и место дисциплины (модуля) в структуре образовательной программы	4
3. Содержание и структура дисциплины (модуля)	5
4. Типы оценочных материалов, показатели и критерии оценивания	10
5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания	14
6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине.	26
7. Методические материалы по освоению дисциплины (модуля).....	35
8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет	37
9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы.....	40

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

Дисциплина Б1.О.06 Алгоритмизация и программирование обеспечивает формирование у обучающихся следующих общепрофессиональных 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	<p><i>ОПК-6.1. 3-1. Знает основные алгоритмические структуры и структуры данных, типовые алгоритмы, используемые при решении экономических задач</i></p> <p><i>ОПК-6.1. У-1. Умеет реализовывать основные типовые алгоритмы решения задач на языке программирования</i></p>

2. Объем и место дисциплины (модуля) в структуре образовательной программы

Общий объем дисциплины:

10 з.е., 360 ак.час.

– очная форма обучения – контактная работа обучающихся с преподавателем по видам учебных занятий: 132 ак.час на контактную работу с преподавателем, из них 64 ак.час на лекции и 64 ак.час на практические занятия, 4 ак.час – консультации. 174 ак. час на самостоятельную работу обучающихся. 54 ак.час – контроль (экзамен, экзамен);

– очно-заочная форма обучения – контактная работа обучающихся с преподавателем по видам учебных занятий: 28 ак.час на контактную работу с преподавателем, из них 12 ак.час на лекции и 12 ак.час на практические занятия, 4 ак.час – консультации. 314 ак. час на самостоятельную работу обучающихся. 18 ак.час – контроль (экзамен, экзамен).

Б1.О.06 Алгоритмизация и программирование реализуется во 2-м и 3-м семестре 1-го курса и 2-го курса, после изучения дисциплин:

- Алгебра
- Экономическая информатика.

3. Содержание и структура дисциплины (модуля)

3.1. Структура дисциплины (модуля)

Очная форма обучения

№ п/п	Наименование тем и (или) разделов	ВСЕ ГО	Объем дисциплины, ак.час											Форма текущего контроля успеваемости, промежуточн ой аттестации	
			Контактная работа обучающихся с преподавателем по видам учебных занятий								Самостоятельная работа				
			Период теоретического обучения				Период промежуточной атте- стации (сессия)								
			Занятия лекционного типа		Занятия семинарского типа		И К	КСР	КЭ	Ка ттэ к	Кон трол ь	СРкр	СРэ к		СР
			Л	ВЛ	ЛР	ПЗ									
2 сем															
Тема 1.	Введение в теорию алгоритмов	25	4	0	0	4	0	0	0	0	0	0	17	Тестирование Опрос	
Тема 2.	Базовые принципы программирования и основные конструкции языка программирования Python	80	20	0	0	20	0	0	0	0	0	0	40	Тестирование Защита отчетов по практическим работам	
Тема 3.	Основные модели структур данных	36	8	0	0	8	0	0	0	0	0	0	20	Опрос	
Промежуточная аттестация		39	0	0	0	0	0	0	2	27	0	10	0	Экзамен	
Итого 2 сем		180	32			32			2	27		10	77		

3 сем														
Тема 4.	Алгоритмы поиска и сорти-	57	10	0	0	10	0	0	0	0	0	0	37	Опрос

	ровки данных.													Контрольная работа
Тема 5.	Введение в теорию графов. Основные алгоритмы на графах.	84	22	0	0	22	0	0	0	0	0	0	40	Тестирование Защита отчетов по практическим работам Контрольная работа
Промежуточная аттестация		39	0	0	0	0	0	0	2	27	0	10	0	Экзамен
Итого 3 сем		180	32			32			2	27		10	77	
Итого		360	64	0	0	64	0	0	4	54	0	20	154	

Очно-заочная форма обучения

№ п/п	Наименование тем и (или) разделов	ВСЕГО	Объем дисциплины, ак.час											Форма текущего контроля успеваемости, промежуточной аттестации	
			Контактная работа обучающихся с преподавателем по видам учебных занятий								Самостоятельная работа				
			Период теоретического обучения				Период промежуточной аттестации (сессия)								
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Ка ттэ к	Конт роль	СРкр	СРэк		СР
			Л	ВЛ	ЛР	ПЗ									
2 сем															
Тема 1.	Введение в теорию алгоритмов	13	2	0	0	2	0	0	0	0	0	0	9	Тестирование Опрос	

Тема 2.	Базовые	24	2	0	0	2	0	0	0	0	0	0	20	Тестирование
---------	---------	----	---	---	---	---	---	---	---	---	---	---	----	--------------

	принципы программирования и основные конструкции языка программирования Python													Защита отчетов по практическим работам
Тема 3.	Основные модели структур данных	14	2	0	0	2	0	0	0	0	0	0	10	Опрос
Промежуточная аттестация		21	0	0	0	0	0	0	2	9	0	10	0	Экзамен
Итого 2 сем		72	6			6			2	9		10	39	
3 сем														
Тема 4.	Алгоритмы поиска и сортировки данных.	104	2	0	0	2	0	0	0	0	0	0	100	Опрос Контрольная работа
Тема 5.	Введение в теорию графов. Основные алгоритмы на графах.	163	4	0	0	4	0	0	0	0	0	0	155	Тестирование Защита отчетов по практическим работам Контрольная работа
Промежуточная аттестация		21	0	0	0	0	0	0	2	9	0	10	0	Экзамен
Итого 3 сем		288	6			6			2	9		10	255	
Итого		360	12	0	0	12	0	0	4	18	0	20	294	

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.

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

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.

3.2. Содержание дисциплины

Тема 1. Введение в теорию алгоритмов. ОПК ОС-6.1

Информационный процесс. Понятие и свойства алгоритма. Базовые алгоритмические структуры. Связь между алгоритмом и понятием функции. Оценка эффективности работы алгоритма. Итерация, рекуррентность и рекурсивность. Инвариант в программировании, верификация программ.

Технические и программные средства реализации информационных процессов. Современные технологии разработки программного обеспечения. Документирование, тестирование и верификация программного кода. Технологии разработки программ.

Тема 2. Базовые принципы программирования и основные конструкции языка программирования Python. ОПК ОС-6.1

Концепция данных и основные конструкции языка программирования Python. Программирование линейных, разветвляющихся, циклических алгоритмов. Технологии работы со структурами данных в Python (списки, строки, кортежи, множества, словари). Вложенные последовательности в Python. Работа с датой и временем в Python. Технологии работы с файлами. Понятие и виды функций в Python. Встроенные функции. Технологии создания пользовательских функций и модулей. Технологии подключения и работы с модулями random, string и др. Работа с датой и временем в Python. Технологии работы с файлами.

Тема 3. Основные модели структур данных. ОПК ОС-6.1

Абстрактные структуры данных: стек, очередь, очередь с приоритетом, ассоциативный массив. Отображение абстрактных структур данных на структуры хранения. Базы данных. Нормирование баз данных. Нормальные формы баз данных. Проектирование баз данных – приведение к нормальным формам.

Тема 4. Алгоритмы поиска и сортировки данных. ОПК ОС-6.1

Сущность алгоритмов сортировки. Классификация алгоритмов сортировки. Реализация алгоритмов сортировки вставками, сортировки выбором и пузырьковой сортировки на языке программирования Python. Быстрая, пирамидальная сортировка и сортировка Шелла: сущность и сферы применения.

Понятие и классификация алгоритмов поиска. Последовательный, бинарный и интерполяционный поиск и их реализация на языке программирования Python.

Использование деревьев в структурах данных, бинарные и квази-бинарные деревья поиска. Оценки алгоритмической сложности операций поиска, добавления и удаления элемента.

Тема 5. Введение в теорию графов. Основные алгоритмы на графах. ОПК ОС-6.1

Понятие и основные элементы графа. Неориентированные и ориентированные графы. Способы задания графа. Маршруты, цепи, циклы. Основные алгоритмы на графах. Основные алгоритмы на графах.

Топологические и экономические сети. Теория экономических сетей. Алгоритмы на ориентированных графах. Сепараторы в графах. Поиск в ширину.

Волновой алгоритм. Алгоритм Прима. Алгоритм Крускала. Реализация алгоритмов на языке программирования Python. Сравнение алгоритмов, особенности реализации и область применимости, оценка их трудоемкости.

Нахождение кратчайших путей на графах. Алгоритм Флойда. Алгоритм Литла. Алгоритм Дейкстры. Реализация алгоритмов на языке программирования Python. Сравнение алгоритмов, особенности реализации и область применимости, оценка их трудоемкости.

Задача Коммивояжера. Реализация алгоритмов на языке программирования Python. Сравнение и оценка трудоемкости алгоритмов и способов решения задачи, особенностей их реализации, область применимости. Применение графов для решения транспортной задачи.

Алгоритм Форда-Фалкерсона. Максимальный поток. Реализация алгоритма на языке программирования Python.

Основные задачи раскраски графов. Методы решения и сфера применения.

Критический путь в графе. Поиск в ширину и глубину на примере продукционной модели. Лексикографический поиск

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 по всем дисциплинам (модулям) образовательной программы составляют 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 откры-

того 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 варианты 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.; 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
Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	<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 варианты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г). 	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)
Задание закрытого типа на установление последовательности	Прочитайте текст и установите последовательность	<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, БВА или 135). 	Ответ считается верным, если правильно указана вся последовательность цифр
Задание комбинированного типа с выбором одного правильного ответа из	Прочитайте текст, выберите правильный ответ и запишите	<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. 	Ответ считается верным, если правильно указана цифра или буква и приведены корректные

предложенных и обоснованием выбора	аргументы, обосновывающие выбор 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).	аргументы, используемые при выборе ответа
Задание открытого типа с развернутым ответом	Прочитайте текст и запишите развернутый обоснованный ответ	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	Ответ считается верным: 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

Традиционная система	Бинарная система	ECTS	
		Для традиционной системы	Для бинарной системы
Отлично	Зачтено	A	P/ Passed
		B	P/ Passed
Хорошо		C	P/ Passed
		D	P/ Passed
Удовлетворительно		E	P/ Passed
Неудовлетворительно	Не зачтено	F	F/Failed

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

5.1. В ходе реализации дисциплины Б1.О.06 Алгоритмизация и программирование используются следующие формы текущего контроля успеваемости обучающихся: тестирование, опрос, контрольная работа, Защита отчетов по практическим работам.

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся:

Тема 1. Введение в теорию алгоритмов. ОПК ОС-6.1

Тестовые задания:

1. Алгоритм – это ...
 - 1) программа на машинном носителе;
 - 2) набор правил, указывающий определенные действия, в результате которых входные данные преобразуются в выходные;
 - 3) вычислительная процедура;
 - 4) псевдокод.
2. Свойство алгоритма выполнять на каждом шаге алгоритма однозначные и независимые от исходных данных действия называется ...
 - 1) детерминированность;
 - 2) результативность;
 - 3) корректность;
 - 4) массовость.
3. Рекурсия представляет собой ...
 - 1) способ задания функций, при котором ее значения для произвольных значений аргументов выражаются через значения определяемой функции для больших значений аргументов;
 - 2) способ задания функций, при котором ее значения для произвольных значений аргументов выражаются через значения определяе-

- мой функции для меньших значений аргументов;
- 3) способ задания функций, при котором ее значения заранее известны;
 - 4) способ задания функций, при котором ее значения определяются с помощью подстановки.
4. Свойство алгоритма, обеспечивающее получение результата через конечное число шагов, - ...
- 1) детерминированность;
 - 2) результативность;
 - 3) корректность;
 - 4) массовость.
5. Рекурсивная процедура – это ...
- 1) прямое или косвенное обращение к себе в процессе выполнения алгоритма;
 - 2) выполняемые в процессе выполнения алгоритма арифметические действия;
 - 3) выбор одной из нескольких альтернатив последовательностей действий в зависимости от исходных данных;
 - 4) способ задания функций, при котором значения аргумента заранее известны
6. Свойство алгоритма, обеспечивающее уверенность в получении правильных результатов для всех допустимых данных при решении определенного круга задач, – ...
- 1) детерминированность;
 - 2) результативность;
 - 3) корректность;
 - 4) массовость.
7. Свойство алгоритма, обеспечивающее возможность его применения для множества исходных данных, - ...
- 1) детерминированность;
 - 2) результативность;
 - 3) корректность;
 - 4) массовость.

Вопросы для опроса:

1. Дайте понятие информационного процесса.
2. Дайте определение алгоритма. Какие способы задания алгоритма вы знаете. Приведите примеры.
3. Что означает свойство «конечность» в отношении алгоритма? Почему важно, чтобы алгоритм имел свойство «конечность»?
4. Как называется свойство, при котором алгоритм даёт один и тот же результат при одних и тех же входных данных?
5. Что обозначает термин «пошаговое выполнение» алгоритма?
6. Какие свойства алгоритма обеспечивают его правильность и эффек-

тивность?

7. Как реализуется свойство алгоритма, заключающееся в том, что алгоритм должен состоять из конкретных действий, следующих в определённом порядке.

8. Что влияет на время выполнения алгоритма?

9. Какие существуют показатели эффективности алгоритмов?

10. Как оценивается временная эффективность (количество выполняемых операций)?

11. Как оценивается пространственная эффективность (объём потребляемой памяти)?

12. Как оценивается применимость алгоритма для решения задачи?

13. Какие методы комплексной оценки эффективности алгоритмов существуют?

14. Как оценивается устойчивость алгоритма к ошибкам определённой группы исходных данных?

15. Как оценивается сходимость алгоритма?

16. Что такое жизненный цикл программного продукта, его основные, вспомогательные и организационные процессы?

17. Какие существуют методы проектирования алгоритмов и программ?

18. Что такое модульное программирование и структурное программирование, в чём их особенности?

19. Какие существуют подходы к разработке программ: нисходящий, восходящий, конструктивный и архитектурный?

20. Какие существуют гибкие методики разработки программного обеспечения?

21. Что такое отладка программ?

22. Какие существуют методы тестирования программ, их виды?

23. Что такое баг и в чём его отличие от ожидаемого и фактического результатов проверок сервиса?

24. Что такое план тестирования и какую информацию он содержит?

25. Какие существуют категории дефектов при тестировании программного обеспечения?

26. Каковы наилучшие методы обеспечения качества программного обеспечения?

27. В чём основное назначение тестовой документации? intuit.ru

Тема 2. Базовые принципы программирования и основные конструкции языка программирования Python. ОПК ОС-6.1

Тестовые задания:

1. Выберите верный вариант позиционного присваивания переменных

1) `>>>x, y, *z = (1, 2, 3, 4);`

2) `>>>x, y, z = (1, 2, 3, 4);`

3) >>>x, *y, *z = (1, 2, 3, 4)

2. Каким будет результат выполнения программного кода

```
>>> x = y = [5, 8]
```

```
>>> y[1] = 100
```

```
>>> x, y
```

1) ([5, 8], [100, 8]);

2) ([5, 100], [5, 100]);

3) ([5, 8], [5, 100])

3. Режим работы, позволяющий форматировать текст программы, сохранять его и запускать по истечении определенного промежутка времени

—

...

1) интерактивный;

2) сценарный;

3) пользовательский

4. Режим работы, позволяющий мгновенно получить результат исполнения программного кода — ...

1) интерактивный;

2) сценарный;

3) пользовательский

5. Что будет выведено на экран в результате выполнения следующей программы?

```
a = 15 // (16 % 7)
b = 34 % a * 5 - 29 % 5 * 2
print(a + b)
```

1) 32

2) 38

3) 29

6. Выберите верное имя переменной

1) speedCar;

2) 7 speed Car;

3) speed car

7. Укажите результат исполнения программного кода на Python

```
a = 82 // 3 ** 2 % 7
print(a)
```

1) 2

2) 1

3) 5.1111111111111086

8. Укажите результат исполнения программного кода на Python

```
a = [13, 56, 17]
a.append([87])
a.extend([45, 67])
print(a)
```

- 1) [13, 56, 17, [87], 45, 67]
- 2) [13, 56, 17, 87, [45, 67]]
- 3) [13, 56, 17, 87, 45, 67]

9. Укажите результат исполнения программного кода на Python

```
a = ['Hello']
a.extend('world')
print(a[1])
```

- 1) world
- 2) w
- 3) Hello

10. Укажите результат исполнения программного кода на Python

```
a = [1, 2, 3, 4]
x = 1
for i in range(len(a)):
    x *= i
print(x)
```

- 1) 120
- 2) 24
- 3) 0

Вопросы для защиты отчетов по практическим работам:

28. Опишите основные типы данных, используемые в Python.
29. Какие функции ввода и вывода языка программирования Python?
30. Какие математические операторы используются в Python?
31. Какие встроенные математические функции в Python?
32. Что такое ключевые слова в Python?
33. Основные конструкции языка программирования.
34. Как реализуются линейные алгоритмы на Python? Какие задачи можно решить с помощью линейных алгоритмов? Как анализировать линейные алгоритмы
35. Как реализуются разветвляющиеся алгоритмы на Python? В чем особенности полной и неполной формы условного оператора? Какие задачи можно решить с помощью разветвляющихся алгоритмов? Как анализировать разветвляющиеся алгоритмы?
36. Как реализуются циклические алгоритмы на Python? В чем отличие в использовании операторов цикла for и while? Какое назначение имеют операторы break и continue для управления работой цикла Для чего используется else в циклических алгоритмах?

37. Какие задачи можно решить с помощью циклических алгоритмов? Например, составить схему алгоритма и написать программу вычисления суммы (произведения) конечного ряда. yaklass.ruCyberForum.ru

38. Что такое список в Python? Как создать пустой список? Какие основные методы работы со списками вы знаете? В чем разница между методами `append()` и `extend()`?

39. Как отсортировать список? Опишите методы `sort()` и `sorted()`.

40. Как объединить два списка?

41. Как удалить элемент из списка? Опишите методы `remove()`, `pop()`, `del`

42. Как проверить наличие элемента в списке? Как получить срез списка?

43. Приведите примеры использования списков в реальных задачах

44. Какие способы создания строк существуют в Python? Как форматировать строки в Python? Опишите f-strings

45. Какие методы работы со строками вы знаете?

46. Как разделить строку на части? Метод `split()`

47. Как объединить строки? Метод `join()`

48. Как изменить регистр строки? Методы `upper()`, `lower()`, `capitalize()`

49. Как найти подстроку в строке?

50. Что такое множество в Python? Как его создать? В чем отличие множества от списка? Когда целесообразно использовать множества вместо списков?

51. Какие операции можно выполнять над множествами?

52. Как объединить множества? Как найти пересечение множеств? Как проверить принадлежность элемента множеству?

53. Как удалить элемент из множества?

54. Что такое словарь в Python? Как его создать? Как создать словарь из двух списков?

55. Как добавить элемент в словарь? Как получить значение по ключу? Как перебрать ключи и значения словаря? В каких случаях лучше использовать словари?

56. Какие методы работы со словарями вы знаете?

57. Как удалить элемент из словаря?

58. Как проверить наличие ключа в словаре?

59. Как объединить два словаря?

60. Что такое функция в Python? Как объявить функцию? Какие особенности функций в Python?

Контрольные задания:

Задание 1.

Составить алгоритм и написать программу вычисления дохода по вкладу и конечной величины вклада. Исходные данные: сумма вклада и срок хра-

нения вклада. Процентная ставка зависит от суммы: если сумма вклада меньше 50000 руб., процентная ставка составляет 6%, если больше, то 7% (процент начисляется в конце срока хранения вклада).

Рекомендуемый вид экранной формы:

Сумма вклада (руб.): _____

Срок хранения вклада (год): _____

Процентная ставка (%): _____

Доход от вклада: _____руб.

Сумма вклада в конце срока хранения: _____руб.

Задание 2.

Построить блок-схему и написать программу вычисления значения функции

$$f(x) = \begin{cases} x^2, & \text{если } x \leq 5 \\ 2x^3 - 0,999, & \text{если } 5 < x \leq 10,25 \\ x, & \text{если } x > 10,25 \end{cases}$$

Подобрать контрольные примеры, позволяющие проверить выполнение алгоритма и работу программы

Задание 3.

Дано натуральное число. Напишите программу, которая определяет, является ли последовательность его цифр при просмотре справа налево упорядоченной по неубыванию.

Задание 4.

Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество слов в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран множество, полученное путем объединения множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 5.

Задание 5.

Интернет-магазин предоставляет услугу экспресс-доставки для части

своих товаров по цене 500 руб за первый товар в заказе и 200 руб. – за все последующие. Напишите функцию, принимающую в качестве единственного параметра количество товаров в заказе и возвращающую общую сумму доставки. В основной программе должны производиться запрос Количества позиций в заказе у пользователя и отображаться на экране сумма доставки

Тема 3. Основные модели структур данных. ОПК ОС-6.1

1. Что такое стек и в каких ситуациях его целесообразно использовать?
2. В чем заключается принцип работы очереди (FIFO)? Приведите пример из реальной жизни.
3. Опишите основные отличия между стеком и очередью.
4. Для каких задач применяется очередь с приоритетом?
5. Что такое ассоциативный массив и как он реализован в Python?
6. Как реализовать очередь с помощью списка? Какие сложности могут возникнуть при реализации стека на основе списка?
7. Какие модули Python используются для работы с приоритетной очередью?
8. Опишите способы создания ассоциативного массива в Python.
9. Приведите пример задачи, решаемой с помощью стека.
10. Как реализовать проверку корректности скобочной последовательности с помощью стека?
11. Опишите алгоритм сортировки с использованием приоритетной очереди.
12. Как реализовать поиск элемента в ассоциативном массиве?
13. Сравните эффективность операций добавления/удаления элементов для разных структур данных.
14. Как оптимизировать работу с очередью при большом количестве элементов?
15. Какие ограничения есть у приоритетной очереди в Python?
16. Как обеспечить безопасность при работе с ассоциативными массивами в многопоточной среде?

Тема 4. Алгоритмы поиска и сортировки данных. ОПК ОС-6.1

Вопросы для опроса:

1. Сущность алгоритмов сортировки. Классификация алгоритмов сортировки
2. Как работает алгоритм Пузырьковой сортировки? Как оптимизировать алгоритм? Как работает алгоритм для сортировки по убыванию
3. Как работает алгоритм Сортировки выбором? Как работает алгоритм для небольших списков? Какую сложность имеет алгоритм и на каких наборах данных его работа наиболее эффективна.

4. Как работает алгоритм Сортировки вставками

5. Как работает алгоритм Быстрой сортировки? Как оптимизировать алгоритм? Как работает алгоритм для больших объёмов данных?

6. Понятие и классификация алгоритмов поиска. Последовательный, бинарный и интерполяционный поиск и их реализация на языке программирования Python.

7. Как используются деревья в структурах данных? Какие виды деревьев поиска Вы знаете? В чем особенности применения каждого из них?

Контрольная работа:

Задание 1.

Составить алгоритм сортировки элементов массива [7, 54, 17, 23, 128, 2, 1] по возрастанию методом пузырьковой сортировки.

Задание 2.

Составить алгоритм сортировки вставками для упорядочивания элементов массива [7, 54, 17, 23, 128, 2, 1] по возрастанию.

Задание 3.

Составить алгоритм сортировки слиянием для упорядочивания элементов массива [7, 17, 47, 53, 128, 210] и [4, 10, 27, 55, 121, 270] по возрастанию.

Задание 4.

Составить алгоритм сортировки элементов массива [47, 54, 2, 23, 128, 32, 1] по убыванию методом пузырьковой сортировки.

Задание 5.

Составить алгоритм сортировки элементов массива [47, 54, 2, 23, 128, 32, 1] по убыванию методом быстрой сортировки.

Тема 5. Введение в теорию графов. Основные алгоритмы на графах. ОПК ОС-6.1

Тестирование:

1. Математический объект, состоящий из множества вершин и множества ребер (дуг), находящихся между собой в некотором отношении, - это ...

- 1) граф;
- 2) автомат;
- 3) алгоритм;
- 4) цепь.

2. Граф, между любыми двумя вершинами которого можно построить маршрут, называется ...

- 1) полным графом;
- 2) связным графом;

- 3) простым графом;
- 4) регулярным графом.

3. Связный оргграф без петель и контуров – это ...

- 1) сетевой график;
- 2) маршрут;
- 3) критический маршрут;
- 4) дерево.

4. Маршрут графа, в котором нет повторяющихся ребер, называется...

- 1) циклом;
- 2) цепью;
- 3) связным маршрутом;
- 4) псевдографом.

5. Цикл, проходящий через каждую вершину графа в точности по одному разу, называется ...

- 1) гамильтоновым циклом;
- 2) эйлеровым циклом;
- 3) простым циклом;
- 4) синхронным циклом

Защита отчетов по практическим работам

1. Дайте понятие и опишите основные элементы графа.

Неориентированные и ориентированные графы.

2. Способы задания графа и представления его в программном коде.

3. Маршруты, цепи, циклы. Основные алгоритмы на графах.

4. Опишите сущность волнового алгоритм. Как он реализуется на Python?

5. В чем сущность алгоритмов Прима, Крускала. Реализация алгоритмов на языке программирования Python. Сравнение алгоритмов, особенности реализации и область применимости, оценка их трудоемкости.

6. Нахождение кратчайших путей на графах.

7. Алгоритм Флойда: сущность, области использования, реализация на Python.

8. Алгоритм Литла: сущность, области использования, реализация на Python.

9. Алгоритм Дейкстры. сущность, области использования, реализация на Python.

10. Задача Коммивояжера. Реализация алгоритмов на языке программирования Python. Сравнение и оценка трудоемкости алгоритмов и способов решения задачи, особенностей их реализации, область применимости.

11. Особенности применения графов для решения транспортной задачи.

12. Алгоритм Форда-Фалкерсона. Максимальный поток. Реализация алгоритма на языке программирования Python.

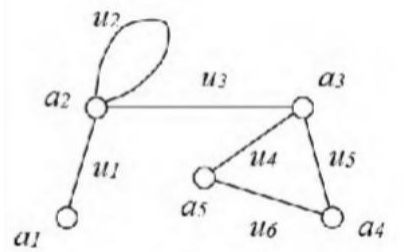
13. Основные задачи раскраски графов. Методы решения и сфера применения.

14. Критический путь в графе. Поиск в ширину и глубину на примере производственной модели. Лексикографический поиск

Контрольная работа:

Задание 1.

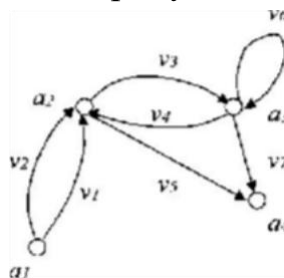
Построить матрицу смежности и матрицу инцидентности для графа



Дать анализ структуры графа с указанием количества вершин, ребер (дуг), петель, циклов, изолированных и висячих вершин графа.

Задание 2.

Построить матрицу смежности и матрицу инцидентности для графа



Дать анализ структуры графа с указанием количества вершин, ребер (дуг), петель, циклов, изолированных и висячих вершин графа.

Задание 3.

Задана матрица смежности графа

☀	0	2	0	0	☿
☿	2	1	1	1	☿
☿	0	1	0	2	☿
☿	0	1	2	0	☿

Требуется:

выяснить, является ли этот граф ориентированным или неориентированным;

построить этот граф;

составить матрицу инцидентности указанного графа.

Задание 4.

Задана матрица инцидентности графа

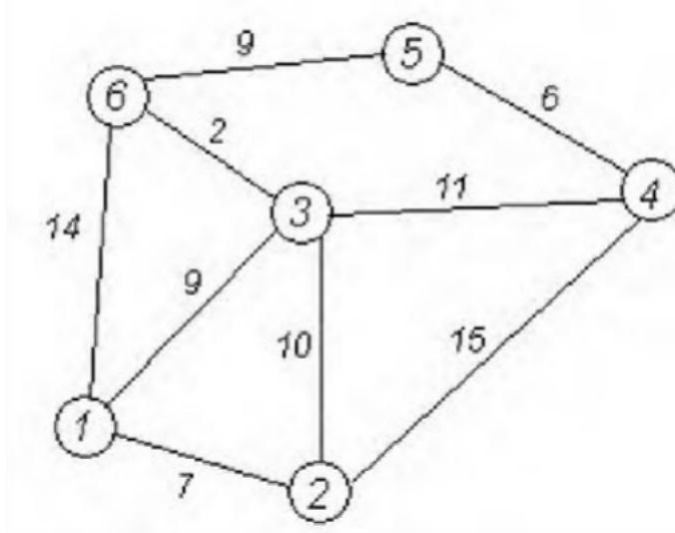
1	0	0	0	0	0
0	0	0	0	0	0
1	1	1	1	0	0
0	0	1	0	1	1
0	0	0	1	0	1

Требуется:

- выяснить, является ли этот граф ориентированным или неориентированным;
- построить этот граф;
- составить матрицу смежности указанного графа

Задание 5.

Для представленного графа найти кратчайшее расстояние от вершины 1 до всех других вершин, программно реализовав алгоритм Дейкстры.



Задание 6.

Используя программу, реализующую алгоритм Форда-Фолкерсона нахождения максимального потока, получите максимальный поток от s к t по данной сети и укажите минимальный разрез, отделяющий s от t . Пропускные способности дуг заданы матрицей.

-	18	16	-	-	9	-
-	-	8	11	7	-	13
-	-	-	-	13	-	19
-	-	10	-	-	15	-
-	-	-	17	-	28	-
-	-	-	-	-	-	14
-	-	-	-	-	-	-

Задание 7.

Сгенерировать обыкновенный граф с взвешенными вершинами. Применить алгоритм поиска в ширину. Результат визуализировать.

Задание 8.

Сгенерировать обыкновенный граф с взвешенными вершинами. Применить алгоритм поиска в глубину. Результат визуализировать.

5.3. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий.

Для решения задач открытого типа студенту разрешается использование специализированного программного обеспечения. При выполнении всех видов работ по дисциплине запрещено использование сервисов искусственного интеллекта.

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

6.1. Промежуточная аттестация (экзамен) проводится в устной форме.

Обучающийся получает экзаменационный билет, включающий два теоретических вопроса и практическое задание. Практическое задание выполняется на компьютере с применением программного обеспечения, используемого при изучении дисциплины. Обучающийся приступает к подготовке, время на подготовку – 40 минут. Обучающийся устно отвечает на теоретические вопросы билета, демонстрируя свои знания и умения. Выполнение практического задания проверяется преподавателем. При необходимости, обучающийся отвечает на дополнительные вопросы преподавателя.

6.2. Типовые оценочные материалы промежуточной аттестации

Типовые проверочные задания для самоподготовки обучающегося к промежуточной аттестации:

Тема 1. Введение в теорию алгоритмов. ОПК ОС-6.1

1. Задания открытого типа.

1.1. Вопросы открытого типа.

№ п.п.	Вопрос
1	Информационный процесс.
2	Понятие алгоритма. Свойства алгоритмов. Применение алгоритмов. Связь между алгоритмом и понятием функции.
3	Итерация, рекуррентность и рекурсивность.
4	Документирование, тестирование и верификация программного кода.
5	Технологии разработки программ. Гибкие технологии разработки. Экстремальное программирование.

6	Асимптотические обозначения в уравнениях. Трудоемкость алгоритмов.
7.	Сложность алгоритма по времени и используемой памяти.
8	Оптимальность реализации алгоритма
9	Назначение тестирования. Функциональное и интеграционное тестирование
10	Верификация. Ревью-кода. Статистический анализ кода.

Тема 2. Базовые принципы программирования и основные конструкции языка программирования Python. ОПК ОС-6.1

1. Задания открытого типа.

1.1. Вопросы открытого типа.

№ п.п.	Вопрос
1	Концепция данных в Python
2	Программирование линейных, разветвляющихся, циклических алгоритмов.
3	Программирование разветвляющихся алгоритмов.
4	Программирование циклических алгоритмов.
5	Технологии работы со структурами данных в Python: списки, строки, кортежи, множества, словари).
6	Технологии работы со строками в Python
7.	Технологии работы со структурами данных в Python: кортежи
8	Технологии работы со структурами данных в Python: множества
9	Технологии работы со структурами данных в Python: словари
10	Вложенные последовательности в Python.
11	Работа с датой и временем в Python.
12	Понятие и виды функций в Python. Встроенные функции. Технологии создания пользовательских функций и модулей.
13	Работа с датой и временем в Python.
14	Технологии работы с файлами.

Тема 3. Основные модели структур данных. ОПК ОС-6.1

1. Задания открытого типа.

1.1. Вопросы открытого типа.

№ п.п.	Вопрос
1	Абстрактные структуры данных: стек, очередь, очередь с приоритетом, ассоциативный массив.
2	Понятие и способы использования стека. Примеры задачи, решаемых с помощью стека.
3	Принцип работы очереди (FIFO). Пример использования структу-

	ры. Опишите основные отличия между стеком и очередью. очередь с приоритетом.
4	Реализация очереди на основе списка. Какие модули Python используются для работы с приоритетной очередью?
5	Сравнительная характеристика структур: стек, очередь, ассоциативный массив
6	Ассоциативный массив: сущность, технологии работы и особенности реализации на Python.
7.	Поиск элемента в ассоциативном массиве
8	Базы данных.
9	Нормирование баз данных.
10	Нормальные формы баз данных. Проектирование баз данных –приведение к нормальным формам.

1.2. Контрольные задания по темам 1-3.

Задание 1.

Построить блок-схему и написать программу вычисления значения функции

$$\begin{aligned}
 & \max\{x^3, \lg(xy)^{cd}\}, \text{ если } \square\square > 3 \\
 & \square = \{3 \min\{\square, \square, \square\square\square\{\square\square, \square\square\}\}, \text{ если } 0 \leq \square\square \leq 3 \\
 & 2^{\square\square} - \square, \text{ если } \square\square < 0
 \end{aligned}$$

Подобрать контрольные примеры, позволяющие проверить выполнение алгоритма и работу программы.

Задание 2.

Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление последнего элемента списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех вещественных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество слов в строке, начинающихся на указанную пользователем букву;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 3.

Задание 3.

На обработку поступает последовательность из 10 целых чисел (каждое на отдельной строке). Нужно написать программу, которая выводит на экран количество неотрицательных чисел последовательности и их произведение. Если неотрицательных чисел нет, требуется вывести на экран «NO» (без кавычек). Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их ровно 4). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

Sample Input:

6
453
455623
25
0
65
112
44
3
90

Sample Output:

10
0

```
1 count = 0
2 p = 0
3 for i in range(1, 10):
4     x = int(input())
5     if x > 0:
6         p = p * x
7         count = count + 1
8 if count > 0:
9     print(x)
10    print(p)
11 else:
12    print('NO')
```

Задание 4.

На обработку поступает натуральное число. Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 3. Если в числе нет цифр, кратных 3, требуется на экран вывести «NO» (без кавычек). Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их ровно 5). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

Sample Input:

46993845

Sample Output:

9

```
1 n = int(input())
2 max_digit = n % 10
3 while n > 0:
4     digit = n % 10
5     if digit % 3 == 0:
6         if digit < max_digit:
7             digit = max_digit
8     n = n % 10
9 if max_digit == 0:
10    print('NO')
11 else:
12    print(max_digit)
```

Задание 5.

Напишите программу, которая осуществляет поиск того элемента массива целых чисел, сумма цифр которого максимальна.

Задание 6.

Напишите программу, которая вычислит сумму первых цифр дробной части тех элементов вещественного массива, которые расположены после минимального его элемента

Задание 7.

Напишите программу сортировки массива целых чисел методом выбора минимального элемента

Задание 8.

Напишите программу формирования **нового** массива целых чисел, в котором сначала располагаются элементы, кратные трём, а затем - все остальные.

Задание 9.

Напишите программу, которая определит, сколько раз в заданной строке встречаются цифровые символы и символы верхнего регистра (A–Z).

Задание 10.

Во входной строке заданы целые положительные числа, означающие цену в копейках. Напишите программу, которая формирует строку, выражающую цену в рублях и копейках.

Тема 4. Алгоритмы поиска и сортировки данных. ОПК ОС-6.1

1. Задания открытого типа.

1.1. Вопросы открытого типа.

№ п.п.	Вопрос
1	Сущность алгоритмов сортировки. Классификация алгоритмов сортировки.
2	Алгоритм сортировки вставками: сущность, примеры использования и реализация на языке программирования Python
3	Сортировка выбором: сущность, примеры использования и реализация на языке программирования Python
4	Алгоритм пузырьковой сортировки: сущность, примеры использования и реализация на языке программирования Python
5	Быстрая сортировка: сущность, примеры использования и реализация на языке программирования Python
6	Пирамидальная сортировка: сущность, примеры использования и реализация на языке программирования Python
7.	Сортировка Шелла: сущность, примеры использования и реализация на языке программирования Python
8	Понятие и классификация алгоритмов поиска.
9	Последовательный, бинарный и интерполяционный поиск и их реализация на языке программирования Python.
10	Использование деревьев в структурах данных, бинарные и квази-бинарные деревья поиска. Оценки алгоритмической сложности операций поиска, добавления и удаления элемента.

Тема 5. Введение в теорию графов. Основные алгоритмы на графах. ОПК ОС-6.1

1. Задания открытого типа.

1.1. Вопросы открытого типа.

№ п.п.	Вопрос
1	Понятие и основные элементы графа. Неориентированные и ориентированные графы. Способы задания графа.
2	Топологические и экономические сети. Теория экономических сетей.
3	Алгоритмы на ориентированных графах. Сепараторы в графах.
4	Волновой алгоритм.
5	Алгоритм Прима. Алгоритм Крускала. Реализация алгоритмов на языке программирования Python. Сравнение алгоритмов, особенности реализации и область применимости, оценка их

	трудоемко-
	сти.
6	Алгоритм Флойда: сущность, области использования, реализация на Python.
7.	Алгоритм Литла: сущность, области использования, реализация на Python.
8	Алгоритм Дейкстры. сущность, области использования, реализация на Python.
9	Задача Коммивояжера. Реализация алгоритмов на языке программирования Python. Сравнение и оценка трудоемкости алгоритмов и способов решения задачи, особенностей их реализации, область применимости.
10	Особенности применения графов для решения транспортной задачи.
11	Алгоритм Форда-Фалкерсона. Максимальный поток. Реализация алгоритма на языке программирования Python.
12	Основные задачи раскраски графов. Методы решения и сфера применения.
13	Критический путь в графе. Поиск в ширину и глубину на примере производственной модели.

1.2. Контрольные задания по темам 4-5.

Задание 1.

Составить алгоритм сортировки элементов массива [120, 5, 17, 23, 12, 25, 13] по возрастанию методом пузырьковой сортировки.

Задание 2.

Составить алгоритм сортировки вставками для упорядочивания элементов массива [25, 13, 2, 7, 25, 0, 4] по возрастанию.

Задание 3.

Составить алгоритм сортировки элементов массива [47, 54, 2, 23, 128, 32, 1] по убыванию методом быстрой сортировки.

Задание 4.

По заданной матрице смежности постройте граф и ответьте на следующие вопросы:

- укажите степени вершин 2 и 4;
- укажите вершины, степень которых равна 3;
- укажите количество и номера четных вершин в графе;
- укажите висячие вершины;

- д) сколько ребер содержит дополнение графа;
- е) укажите вершины, смежные относительно вершины б;

ж) из заданного графа удалили вершину 2. Сколько ребер в получившемся подграфе?

	1	2	3	4	5	6	7	8
1		1						
2	1		1		1	1	1	1
3		1				1	1	
4								1
5		1						
6		1	1					1
7		1	1					1
8		1		1		1	1	

Задание 5.

По заданной матрице инцидентности постройте граф и ответьте на следующие вопросы:

- з) укажите количество ребер, инцидентных вершине 3? Вершине 5?
- и) укажите вершины, степень которых равна 3;
- к) укажите номера висячих вершин;
- л) укажите номера четных вершин;
- м) сколько ребер содержит дополнение графа;
- н) сколько в дополнении графа ребер, инцидентных вершине 2?

Вершине 6?

1	1									
2	1	1	1	1	1	1				
3		1					1			
4								1		
5			1							
6				1			1			1
7					1				1	
8						1		1	1	1

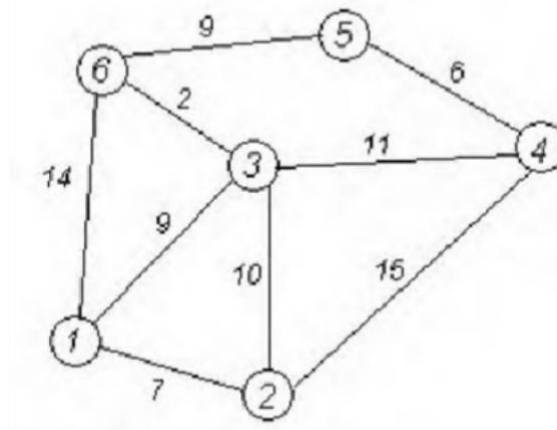
Задание 6.

Используя программу, реализующую алгоритм Форда-Фолкерсона нахождения максимального потока, получите максимальный поток от s к t по данной сети и укажите минимальный разрез, отделяющий s от t . Пропускные способности дуг заданы матрицей.

$$\begin{pmatrix} - & 9 & - & 11 & - & 17 & - \\ - & - & 6 & - & 8 & - & 12 \\ - & - & - & - & - & - & 7 \\ - & 5 & - & - & - & 5 & 4 \\ - & - & - & - & - & 7 & - \\ - & - & - & - & - & - & 9 \\ - & - & - & - & - & - & - \end{pmatrix}.$$

Задание 7.

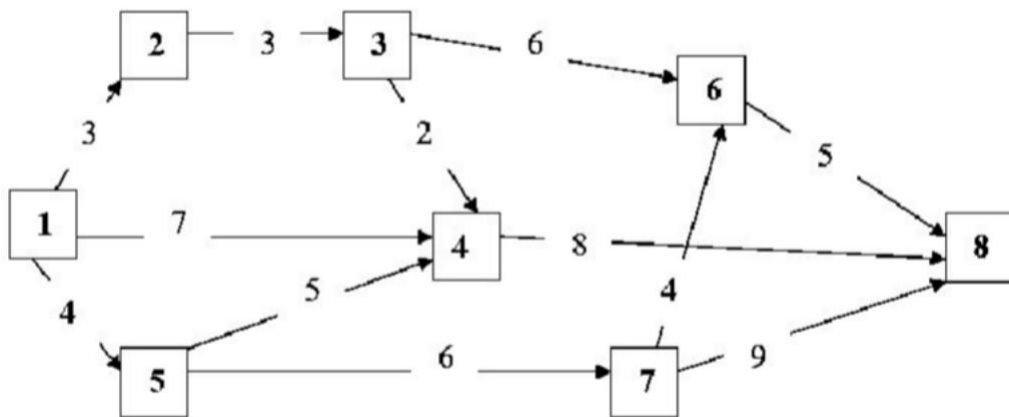
Для представленного графа найти кратчайшее расстояние от вершины 1 до всех других вершин, программно реализовав алгоритм Дейкстры.



Здание 8.

Дан сетевой график выполнения проекта с нанесенными на него длительностями d_{ij} отдельных работ. Требуется:

- 1) Построить критический путь $L_{крит}$ и определить его длительность.
- 2) Построить все полные пути L для этого графика, определить их длительности $T(L)$ и резервы времени $R(L)$.
- 3) Для каждого события $\{ j \}$ сетевого графика рассчитать максимальный $\langle B_j \rangle$ и минимальный $[b_j]$ индексы
- 4) Нанести на сетевой график результаты проведенного выше анализа.
- 5) Определить величины полного $R_n(i-j)$, свободного $R_c(i-j)$ и независимого $R_n(i-j)$ резервов времени для следующих работ: 2-3 и 4-8.

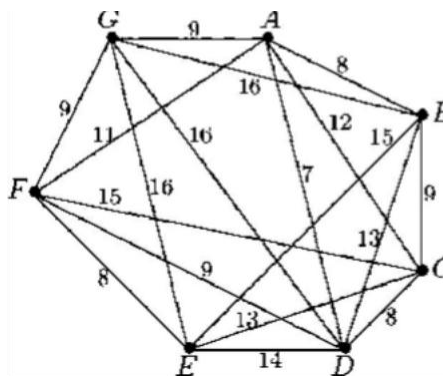


Задание 9.

Сгенерировать обыкновенный граф с взвешенными вершинами. Применить алгоритм поиска в глубину. Результат визуализировать.

Задание 10.

Дан граф с N вершинами (городами) и взвешенными рёбрами (путями между городами с определённой длиной или стоимостью). Требуется найти гамильтонов цикл минимальной длины — такой маршрут, который проходит через каждую вершину ровно один раз и возвращается в начальную точку, при этом минимизируя суммарный вес (расстояние).



6.3. Критерии и шкала оценивания.

КРИТЕРИИ ОЦЕНИВАНИЯ	РЕЗУЛЬТАТ
Дан полный развернутый ответ на теоретические вопросы билета, обучающийся продемонстрировал знания материала в объеме изученной учебной дисциплины, уверенно и исчерпывающе ответил на дополнительные вопросы, сопровождая ответ примерами по проблематике поставленного вопроса, без ошибок выполнил предложенное в билете практическое задание	отлично
Дан развернутый ответ на теоретические вопросы билета, продемонстрированы знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу. На дополнительные вопросы даны аргументированные ответы, приведены примеры по проблематике поставленного вопроса. Однако при ответе и/или выполнении практического задания допущены несущественные неточности.	хорошо
Дан ответ, свидетельствующий о знании основных процессов изучаемой дисциплины, но отличающийся недостаточной глубиной и полнотой, недостаточным умением давать аргументированные ответы и приводить примеры. Допущено несколько ошибок в содержании ответа и решении практического задания.	удовлетворительно
Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е. обучающийся не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.	неудовлетворительно

6.4. Для решения контрольных заданий обучающийся использует компьютер с соответствующим программным обеспечением

7. Методические материалы по освоению дисциплины (модуля)

Обучение по дисциплине предполагает изучение курса на аудиторных

(лекционных, практических) занятиях, самостоятельную работу обучающихся-

ся и проведение контрольных мероприятий (текущей и промежуточной аттестации).

В начале учебного курса рекомендуется ознакомиться с рабочей программой дисциплины, обратив внимание на следующие ключевые моменты:

- перечень знаний, умений и навыков, которыми обучающийся должен овладеть в результате освоения дисциплины;
- тематический план проведения занятий;
- формы и материалы для проведения текущего контроля;
- учебники и учебные пособия по дисциплине, а также электронными ресурсами;
- оценочные материалы для проведения текущей и промежуточной аттестации.

С целью обеспечения успешного освоения курса обучающийся должен готовиться к лекционным и практическим занятиям.

На лекционных занятиях рекомендуется конспектировать основные положения излагаемого преподавателем материала. При подготовке к занятиям целесообразно самостоятельно прорабатывать лекционный материал, особое внимание следует уделять возникающим при этом вопросам, выделяя или выписывая указанные моменты для дальнейшего уточнения и обсуждения на последующих занятиях. В случае необходимости обращаться к преподавателю за консультацией. Таким образом, подготовка к лекции заключается в:

- внимательном прочтении материала предыдущей лекции;
- знакомстве с темой предстоящей лекции (по тематическому плану или информации лектора);
- предварительном знакомстве с учебным материалом по учебнику и учебным пособиям;
- уяснении места изучаемой темы в своей профессиональной подготовке;
- записи возможных вопросов преподавателю.

Практические занятия предполагают их проведение в различных формах (в том числе с использованием информационно-коммуникационных технологий) с целью систематизации, углубления и конкретизации полученных теоретических знаний, а также формирования навыков их применения при решении практических задач. Подготовка к практическим занятиям заключается в:

внимательном прочтении теоретического материала лекций, относящихся к данному занятию, изучении учебного материала в рекомендуемых учебно-методических пособиях;

ответе на приведенные в рабочей программе дисциплины вопросы для самостоятельной подготовки к занятиям лекционного, практического (семинарского) типов, лабораторным работам;

выполнении заданий для самостоятельной работы по соответствующей теме.

В рамках практических занятий предусмотрено проведение мероприя-

тий текущего контроля с целью выявления уровня освоения полученных знаний, умений, навыков и компетенций. Для осуществления текущего контроля в рабочей программе предусмотрены различные формы. При подготовке к контрольным мероприятиям рекомендуется ознакомиться с представленными в рабочей программе типовыми материалами для проведения текущего контроля успеваемости обучающихся.

Изучение дисциплины предполагает самостоятельное освоение обучающимся части учебного материала в соответствии с перечнем вопросов для самостоятельного изучения.

Перечень вопросов для самостоятельного изучения:

1. Эволюция и классификация языков программирования
2. Компиляторы и интегрированные среда разработки
3. Функциональные возможности интегрированных сред программирования
4. Обзор современных технологий разработки программного обеспечения
5. Структурная организация данных
6. Стандартизация оформления графической формы представления алгоритма (ГОСТ 19.701-90 «ЕСПД. Схемы алгоритмов программ, данных и систем. Условные обозначения и правила выполнения»)
7. Алгоритм сортировки Шелла
8. Пирамидальная сортировка

При этом особое внимание следует обратить на самостоятельное изучение рекомендованной учебно-методической литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем.

Подготовка к промежуточной аттестации. предполагает целенаправленную, регулярную, систематическую работу при изучении дисциплины в рамках аудиторных занятий и самостоятельной работы. Систематическое выполнение учебной работы на лекциях и практических занятиях, а также в рамках самостоятельной работы позволит успешно освоить дисциплину и создать хорошую базу для успешного прохождения промежуточной аттестации. Непосредственно подготовку к процедуре аттестации рекомендуется осуществлять в соответствии с приведенным в рабочей программе перечнем типовых оценочных материалов для проведения промежуточной аттестации.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Алгоритмизация и программирование: учебник для вузов / Д. А. Бархатова, Н. И. Пак, А. А. Петрова, И. А. Яшина. – Санкт-Петербург: Лань, 2025. – 348 с. – ISBN 978-5-507-52244-6. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/482915>. – Режим доступа: для авториз. пользователей.
2. Гниденко, И.Г. Технологии и методы программирования: учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2025. – 241 с. – (Высшее образование). – ISBN 978-5-534-18130-2. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/581329>.
3. Крупский, В.Н. Теория алгоритмов. Введение в сложность вычислений: учебник для вузов / В.Н. Крупский. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2025. – 91 с. – (Высшее образование). – ISBN 978-5-534-21288-4. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/563580>.
4. Трофимов, В.В. Алгоритмизация и программирование: учебник для вузов / В.В. Трофимов, Т.А. Павловская. – 4-е изд. – Москва: Издательство Юрайт, 2025. – 108 с. – (Высшее образование). – ISBN 978-5-534-20430-8. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/562040>.
5. Федоров, Д.Ю. Программирование на python: учебное пособие для вузов / Д.Ю. Федоров. – 6-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2025. – 187 с. – (Высшее образование). – ISBN 978-5-534-19666-5. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/556864>.
6. Федоров, Д.Ю. Программирование на python: учебное пособие для вузов / Д.Ю. Федоров. – 6-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2025. – 187 с. – (Высшее образование). – ISBN 978-5-534-19666-5. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/556864>.
7. Черпаков, И.В. Основы программирования: учебник и практикум для вузов / И.В. Черпаков. – 2-е изд. – Москва: Издательство Юрайт, 2025. – 196 с. – (Высшее образование). – ISBN 978-5-534-18759-5. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/560807>.

8.2. Дополнительная литература

1. Асанов, М.О. Дискретная математика: графы, матроиды, алгоритмы: учебное пособие для вузов / М.О. Асанов, В.А. Баранский, В.В. Расин. – 4-е изд., стер. – Санкт-Петербург: Лань, 2024. – 364 с. – ISBN 978-5-507-47699-2. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/407519>. – Режим доступа: для авториз. поль-

зователей.

2. Дискретная математика: прикладные задачи и сложность алгоритмов: учебник и практикум для вузов / А.Е. Андреев, А.А. Болотов, К. В. Ко-ляда, А.Б. Фролов. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2025. – 317 с. – (Высшее образование). – ISBN 978-5-534-04246-7. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/563029>.

3. Иванов, Б.Н. Дискретная математика и теория графов: учебник для вузов / Б.Н. Иванов. – Москва: Издательство Юрайт, 2025. – 177 с. – (Высшее образование). – ISBN 978-5-534-14470-3. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/567929>.

4. Информационные технологии: учебник для вузов / В.В. Трофимов, О.П. Ильина, В.И. Кияев, Е.В. Трофимова; под редакцией В. В. Трофимова. – Москва: Издательство Юрайт, 2025. – 546 с. – (Высшее образование). – ISBN 978-5-534-18340-5. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/568880>.

8.3. Нормативные правовые документы и иная правовая информация

Национальный проект «Экономика данных и цифровая трансформация государства» (утв. протоколом заседания президиума Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам от 20 декабря 2024 г. № 12 пр). – URL: <http://government.ru/rugovclassifier/923/events/>

Указ Президента Российской Федерации от 09.05.2017 г. № 203 «О Стратегии развития информационного общества в Российской Федерации на 2017-2030 годы» – URL: <http://www.kremlin.ru/acts/bank/41919>

8.4. Интернет-ресурсы

Документация по языку программирования Python – <https://pydocs.ru/home/>

Документация по языку Python3 <https://docs-python.ru/>

ЭБС «Лань» – <https://lib.ranepa.ru/ru/informatsionnye-resursy/russkoyazychnye-resursy/45-ebs-izdatelstva-lan>

ЭБС «IPR SMART» - <https://lib.ranepa.ru/ru/informatsionnye-resursy/russkoyazychnye-resursy/32-ebs-iprbooks>

ЭБС «Znanium» - <https://lib.ranepa.ru/ru/informatsionnye-resursy/russkoyazychnye-resursy/389-znanium>

ЭБС «ibooks.ru» - <https://lib.ranepa.ru/ru/informatsionnye-resursy/russkoyazychnye-resursy/681-ebs-ibooks-ru>

ЭБС издательства «Юрайт» - <https://lib.ranepa.ru/ru/informatsionnye-resursy/russkoyazychnye-resursy/55-ebs-yurajt>

Справочно-правовая система «Консультант Плюс» – <http://www.consultant.ru/>

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

Требования к аудитории:

Лекционные

Семинарские

Помещения для самостоятельной работы

Требования к оборудованию:

Доска

Проектор

Компьютер (стационарный) или ноутбук

• Компьютеры с установленным программным обеспечением

Требования к программному обеспечению:

Наименование программного обеспечения	Назначение
7-Zip	Архиватор с высокой степенью сжатия. Свободная лицензия
Adobe Acrobat	Чтение, печать и рецензирование файлов PDF. Свободная лицензия
Adobe reader	Просмотр, печать и внесение аннотаций в документы PDF. Свободная лицензия
Anaconda	дистрибутив языков программирования Python и R с открытым исходным кодом, разработанный для научных вычислений, науки о данных и машинного обучения
Draw.io	бесплатный онлайн-сервис для создания диаграмм, блок-схем и визуальных схем
Kaspersky Endpoint Security	Антивирусное ПО
MS Office	Пакет офисных приложений.
MS Visual Studio 2022	интегрированная среда разработки (IDE), созданная компанией Microsoft. Предназначена для разработки программного обеспечения на различных языках программирования, включая C#, VB.NET, C++, Python и другие
Mts-link Telemost	Экосистема сервисов для бизнес-коммуникаций и совместной работы.
Open Office	Свободный пакет офисных приложений. Свободная лицензия
Yandex Browser	Браузер Свободная лицензия
Yandex Messenger	приложение для общения, групповых чатов, а также аудио – и видеозвонков
Яндекс Диск	облачное хранилище для файлов, данных и видео от Яндекс
Яндекс Календарь	бесплатный персональный информационный менеджер