

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлутков
Должность: директор
Дата подписания: 14.08.2026 15:21:14
Уникальный программный ключ:
880f7c07c583b07b775f6604c39281b15c9f12

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет среднего профессионального образования

УТВЕРЖДЕНА
решением цикловой (методической)
комиссии общепрофессиональных
дисциплин и по профессиональным
модулям специальности 09.02.07
Информационные системы и
программирование
Протокол от 31.10.2025 № 2

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОП.06 Основы алгоритмизации и программирования

Специальность – 09.02.11 Разработка и управление программным обеспечением

Профиль – на базе основного общего образования

Квалификация – программист

Форма обучения – очная

Год набора – 2026

Санкт-Петербург 2025 год

Автор-составитель: Генрихов Василий Сергеевич, преподаватель ФСПО СЗИУ РАНХиГС.

СОДЕРЖАНИЕ

1. Общие положения	4
1.1. Область применения программы	4
1.2. Место дисциплины в структуре основной профессиональной образовательной программы	4
1.3. Цели и задачи учебной дисциплины	4
1.4. Планируемые результаты обучения по дисциплине	4
2. Структура и содержание дисциплины	12
2.1. Объем учебной дисциплины и виды работ	12
2.2. Тематический план и содержание дисциплины	12
2.3. Регламент распределения видов работ по дисциплине с ДОТ	18
3. Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся	19
3.1. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации.....	19
3.2. Оценочные средства текущего контроля успеваемости обучающихся	21
3.3. Оценочные средства промежуточной аттестации обучающихся	22
4. Методические указания для обучающихся по освоению дисциплины	25
5. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»	28
6. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы	33

1 Общие положения

1.1 Область применения программы

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» является частью профессиональной подготовки обучающихся направления подготовки 09.02.11 «Разработка и управление программным обеспечением».

1.2 Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» является частью профессиональной подготовки и входит в профессиональный цикл дисциплин.

Для успешного освоения дисциплины предполагается освоение таких предшествующих дисциплин, как «Информатика», «Математический аппарат в отрасли информационных технологий», «Основы работы с информацией», «Разработка программных модулей», «Операционные системы и среды». Полученные в результате освоения дисциплины знания, умения и практические навыки необходимы и будут полезны для последующего изучения дисциплины «Инструментальные средства разработки программного обеспечения».

Дисциплина изучается на 2 курсе в 4 семестре.

1.3 Цели и задачи учебной дисциплины

Цель дисциплины «Основы алгоритмизации и программирования» - формирование навыков работы в среде программирования, разработки алгоритмов для решения конкретных задач, реализации готовых и разработанных алгоритмов на выбранном языке программирования.

Дисциплина «Основы алгоритмизации и программирования» включена в обязательную часть общепрофессионального цикла образовательной программы.

1.4 Планируемые результаты обучения по дисциплине

Перечень общих компетенций и компетенций

Код ПК, ОК	Умения	Знания	Владеть навыками
ОК.01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	– распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части	– актуальный профессиональный и социальный контекст, в котором приходится работать и жить	-

ОК.02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации	– номенклатура информационных источников, применяемых в профессиональной деятельности	-
ОК.0303 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	– применять современную научную профессиональную терминологию	– современная научная и профессиональная терминология	-
ОК.04 Эффективно взаимодействовать и работать в коллективе и команде	– взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности	– психологические особенности личности	-
ОК.05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	– грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке	– правила оформления документов	-
ОК 06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе	– демонстрировать осознанное поведение	– традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений	-

традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения			
ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	– соблюдать нормы экологической безопасности	– правила экологической безопасности при ведении профессиональной деятельности	-
ОК 08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;	– пользоваться средствами профилактики перенапряжения, характерными для данной специальности	– средства профилактики перенапряжения	-
ОК.09 Пользоваться профессиональной документацией на государственном и иностранном языках.	– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы	– правила построения простых и сложных предложений на профессиональные темы	-

ПК 2.2 «Разрабатывать модули программного обеспечения»	– разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий – применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей – анализировать требования и определять функциональность модуля – создавать интерфейсы для взаимодействия с другими модулями и системами – обеспечивать безопасность, производительность и масштабируемость при разработке модулей – оптимизировать проектируемые модули для повышения их эффективности и качества – работать с системой контроля версий – улучшать производительность модулей, выявляя и устраняя узкие места – проводить анализ и мониторинг производительности приложений – применять инструменты для рефакторинга и оптимизации программного кода	– язык программирования, основные конструкции, синтаксис – паттерны проектирования – структуры данных – принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP – работа с инструментальным программным обеспечением – методы оптимизации кода и алгоритмов – эффективные алгоритмы и структуры данных для повышения производительности – многопоточность в программных модулях – методы оптимизации сетевых протоколов для ускорения обмена данными – кэширование данных – управление памятью – техники повышения производительности программного обеспечения	– создание модулей программного обеспечения на различных языках программирования – отладки и тестирования разработанных модулей – применение структурного и объектно-ориентированного программирования – оптимизации кода и алгоритмов программных модулей для увеличения производительности – мониторинга и анализа производительности приложений
---	---	---	--

ПК 2.4 Выполнять тестирование и отладку программного обеспечения	– анализировать требования к программному обеспечению и составлять планы тестирования. – создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям. – выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования. – анализировать результаты тестирования и документировать найденные ошибки. – разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении. – выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования – использовать системы контроля дефектов ПО – составлять отчет о выполнении тестирования ПО	– принципы и методы тестирования программного обеспечения. – основы программирования и архитектуры программного обеспечения. – основы баз данных и SQL-запросов. – инструменты для автоматизации тестирования – основы разработки и отладки программного обеспечения на разных языках – программирования – понятие дефекта программного обеспечения – критерии качества ПО – виды и типы тестирования ПО – техники ручного тестирования – техники автоматизированного тестирования – жизненный цикл дефекта ПО – принципы работы в системе контроля дефектов – основные понятия о качестве ПО	– отладки программного обеспечения на уровне программных модулей – тестирования программного обеспечения – формирования тестовых сценариев – подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости) – оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения – настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции – формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами – выполнения тестовых процедур на тестовых данных
---	---	--	--

2 Структура и содержание дисциплины

2.1 Объем учебной дисциплины и виды работ

Виды учебной работы	Объем учебной работы, час.
Учебная нагрузка обучающихся всего, в том числе:	252
лекции	24
практические занятия	158
курсовая работа	-
самостоятельная работа обучающихся	60
консультации	4
промежуточная аттестация	6
Форма промежуточной аттестации	Другая форма контроля, экзамен

2.2 Тематический план и содержание дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенции, формированию которых способствует элемент программы	
Тема 1. Основы алгоритмизации	Содержание учебного материала	4	ОК 01	ПЗ, Т
	Введение в дисциплину. Основные понятия. Определения		ОК 02	
	Алгоритм. Свойства алгоритма. Способы представления алгоритма		ОК03	
	Базовые алгоритмические структуры. Принципы и правила построения блок-схем		ОК 04	
	В том числе практических занятий и лабораторных работ	38	ОК 05	
	Разработка блок-схем линейных алгоритмов		ОК 06	
	Разработка блок-схем разветвляющихся алгоритмов		ОК 07	
	Разработка блок-схем циклических алгоритмов		ОК 08	
	Самостоятельная работа обучающихся	20	ОК 09	
			ПК 2.2, ПК 2.4	
Тема 2. Основы программирования	Содержание учебного материала	10	ОК 01 ОК 02 ОК03 ОК 04 ОК 05 ОК 06 ОК 07 ОК 08 ОК 09 ПК 2.2, ПК 2.4	О, Т, ПЗ
	Базовый синтаксис Python. Разработка линейных программ			
	Типы данных. Булевый тип данных			
	Организация ветвления в программе. Условные операторы			
	Цикл с постусловием. Цикл с предусловием. Цикл с параметром.			
	Вложенные циклы			
	Введение в списки. Методы работы со списками			
	Алгоритмы поиска и сортировки списков			
	Функции. Функции с параметрами. Функции, возвращающие значения			
	Вложенные списки и матрицы			
	Картежи и словари			
	Дополнительные модули			
	В том числе практических занятий и лабораторных работ	60		
	Разработка линейных программ			
	Разработка программ с условными операторами			
	Вложенные и каскадные условные операторы в программе			

	Разработка циклических программ			
	Вложенные циклы			
	Разработка программ со списками			
	Разработка программ с использованием функций			
	Алгоритмы поиска и сортировки списков			
	Разработка программ с использованием вложенных списков			
	Обработка и методы работы с картами и словарями			
	Использование дополнительных модулей в программе			
	Самостоятельная работа обучающихся			
Тема 3. Основы объектно-ориентированного программирования	Содержание учебного материала	10	ОК 01 ОК 02 ОК03 ОК 04 ОК 05 ОК 06 ОК 07 ОК 08 ОК 09 ПК 2.2, ПК 2.4	О, Т, ПЗ
	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.			
	Основные принципы ООП: инкапсуляция, наследование, полиморфизм			
	Классы объектов. Компоненты и их свойства			
	Событийно-управляемая модель программирования. Компонентно-ориентированный подход			
	Этапы разработки приложений			
	Иерархия классов			
	В том числе практических занятий и лабораторных работ			
	Передача имен функций в качестве параметров	60		
	Вложенные классы. Конструкторы. перегрузка			
	Множественное наследование. Полиморфизм			
	Самостоятельная работа обучающихся	10		
	Консультации	4		
	Промежуточная аттестация	6		
Всего:		252		

2.3 Регламент распределения видов работ по дисциплине с ДОТ

Данная дисциплина реализуется с применением дистанционных образовательных технологий (ДОТ). Распределение видов учебной работы, форматов текущего контроля представлены в Таблице 2.3.

Таблица 2.3. — Распределение видов учебной работы и текущей аттестации

Вид учебной работы	Формат проведения
Лекционные занятия	Частично с применением ДОТ
Практические занятия	Частично с применением ДОТ
Текущий контроль	Частично с применением ДОТ
Промежуточная аттестация	Контактная аудиторная работа
Формы текущего контроля	Формат проведения
Тестирование	Частично с применением ДОТ
Опрос	Контактная аудиторная работа
Практические задания	Частично с применением ДОТ

Доступ к системе дистанционных образовательных программ осуществляется каждым обучающимся самостоятельно с любого устройства на портале: <https://sziu-de.ranepa.ru> в соответствии с их индивидуальным паролем и логином к личному кабинету / профилю.

Текущий контроль, проводимый в системе дистанционного обучения, оцениваются как в системе дистанционного обучения, так и преподавателем вне системы. Промежуточный контроль в виде экзамена проводится устно.

Доступ к материалам лекций предоставляется в течение всего семестра по мере прохождения освоения программы. Доступ к каждому виду работ и количество попыток на выполнение задания предоставляется ограниченное время согласно регламенту дисциплины, опубликованному в системе дистанционного обучения. Преподаватель оценивает выполненные обучающимися работы не позднее 14 рабочих дней после окончания срока выполнения.

3 Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся

3.1 Формы и методы текущего контроля успеваемости и промежуточной аттестации обучающихся

Формы текущего контроля успеваемости:

Опрос (О) позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления.

Тестирование (Т) – письменные задания, с вариантами ответов.

Практическое задание (ПЗ) используется для закрепления теоретических знаний и отработки навыков и умений, способности применять знания при решении конкретных задач.

Критерии оценивания текущего и промежуточного контроля аттестации:

Оценки «отлично» заслуживает студент, обнаруживший глубокое знание материала, умение свободно выполнять задания, понимающий взаимосвязь основных понятий темы; при выполнении теста правильно 90% вопросов теста;

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала; успешно выполняющий предусмотренные задания; и допустивший незначительные ошибки: неточность фактов, стилистические ошибки; при выполнении теста правильно даны ответы от 89%-75% вопросов теста.

Оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного материала в объеме, необходимом для дальнейшего изучения дисциплины. Справляющийся с выполнением заданий; допустивший погрешности в ответе, но обладающий необходимыми знаниями для их устранения под руководством преподавателя; при выполнении теста правильно даны ответы от 74%-50% вопросов теста.

Оценки «неудовлетворительно» заслуживает студент, обнаруживший существенные пробелы в знании основного материала; не справляющийся с выполнением заданий, допустивший серьезные погрешности в ответах, нуждающийся в повторении основных разделов курса под руководством преподавателя; при выполнении теста правильно даны ответы от 49% и меньше.

Формы текущего контроля

№п/п	Название темы	Формы текущего контроля успеваемости, промежуточной аттестации
1	Тема 1. Основы алгоритмизации	ПЗ, Т
2	Тема 2. Основы программирования	Т, ПЗ, О
3	Тема 3. Основы объектно-ориентированного программирования	Т, ПЗ, О

Примечание. В столбце «Форма текущего контроля успеваемости, промежуточной аттестации» перечисляются все используемые в учебном процессе по данной дисциплине формы контроля освоения материала. (Т – тестирование; ПЗ – практическое задание, О- устный опрос).

3.2 Оценочные средства текущего контроля успеваемости обучающихся

Вопросы к устному опросу:

Тема 1. Основы алгоритмизации

1. Дать определение алгоритма.
2. Сделать обзор свойств алгоритма.
3. Перечислить способы описания алгоритма.
4. Стандарты описания блок-схем.
5. Дать определение компилятору и интерпретатору.
6. Перечислить основные алгоритмические структуры.
7. Привести примеры структур ветвления.
8. Сделать обзор циклических структур.
9. Дать определение арифметическому циклу.
10. Рассмотреть операторы цикла с предусловием
11. Рассмотреть операторы цикла с предусловием
12. Дать определение итерации
13. Дать определение рекурсии

Тема 2. Основы программирования

1. Дать определение переменной.
2. Сделать обзор типов данных.
3. Дать определение областям видимости переменных.
4. Сделать обзор функций преобразования типов.
5. Перечислить операторы условия, привести пример программы с условием
6. Вложенные и каскадные условия
7. Дать определение циклу с условием. Назвать соответствующий оператор Python и его особенности
8. Цикл с предусловием. Определение и оператор Python. Особенности использования.
9. Опишите назначение операторов break и continue. Приведите пример применения.
10. Сделать обзор вложенных циклов
11. Айте определение спискам как типу данных Python. Опишите методы и особенности

работы со списками

12. Индексация и срезы. Понятия и примеры.
13. Перечислите и дайте характеристику алгоритмам поиска и сортировки, которые вы знаете.
14. Опишите функции в Python. Какие виды функций бывают?
15. Проведите обзор картежей и словарей.
16. Перечислите дополнительные модули и библиотеки, которые вы знаете. Опишите их назначение.

Тема 3. Основы объектно-ориентированного программирования

1. Определите базовые понятия ООП.
2. Перечислите свойства и методы объектов
3. Перечислите основные принципы ООП
4. Опишите компоненты и свойства классов
5. Охарактеризуйте конструкторы и деструкторы классов
6. Метод: понятие, правила записи. Вызов метода. Передача параметров по значению.
7. Доступ к классам и объектам. Модификаторы доступа
8. Инкапсуляция как управление доступом к данным.
9. Дайте характеристику событийно-управляемой модели программирования
10. Что такое компонентно-ориентированный подход
11. Охарактеризуйте этапы разработки приложений
12. Дайте характеристику Иерархии классов
13. Проведите обзор наследования
14. Определите понятие Полиморфизм
15. Что такое перегрузка?

Примеры типовых заданий для практических работ

Тема 1. Основы алгоритмизации

Практическая работа 1. Разработка линейных алгоритмов

1. Перечислить свойства алгоритма.
2. Составить линейный алгоритм на естественном языке
3. Составить блок-схему алгоритма.
5. Оформить отчет о проделанной работе

Практическая работа 2. Разработка разветвляющихся алгоритмов

1. Перечислить свойства алгоритма.
2. Составить алгоритм с односторонним ветвлением на естественном языке
3. Составить блок-схему алгоритма.
4. Составить алгоритм с двухсторонним ветвлением на естественном языке
5. Составить блок-схему алгоритма.
6. Составить алгоритм с вложенным условием на естественном языке
7. Составить блок-схему алгоритма.
8. Оформить отчет о проделанной работе

Практическая работа 3. Разработка циклических алгоритмов

1. Перечислить свойства алгоритма.
2. Составить алгоритм для цикла с параметром на естественном языке
3. Составить блок-схему алгоритма.
4. Составить алгоритм для цикла с предусловием на естественном языке

5. Составить блок-схему алгоритма.
6. Оформить отчет о проделанной работе

Тема 2. Основы программирования

Практическая работа 4. Разработка линейных программ

1. На вход программе подаются два целых числа a и b . Напишите программу, которая выводит:
 - сумму чисел a и b ;
 - разность чисел a и b ;
 - произведение чисел a и b ;
 - частное чисел a и b ;
 - целую часть от деления числа a на b ;
 - остаток от деления числа a на b ;
 - корень квадратный из суммы их 10-х степеней: $\sqrt{a^{10} + b^{10}}$
2. Напишите программу для вычисления индекса массы тела (ИМТ) человека. ИМТ показывает весит человек больше или меньше нормы для своего роста. ИМТ человека рассчитывают по формуле:

$$\text{ИМТ} = \frac{\text{масса (кг)}}{\text{рост(м)} \times \text{рост(м)}},$$

На вход программе подается два числа: масса и рост человека, каждое на отдельной строке. Все входные числа являются вещественными, используйте для них тип данных `float`.

3. Дана строка текста. Напишите программу для подсчета стоимости строки, исходя из того, что один любой символ (в том числе пробел) стоит 60 копеек. Ответ дайте в рублях и копейках в формате: `XX руб. XX коп.`
4. Дано пятизначное натуральное число. Напишите программу, которая изменит порядок его цифр на обратный.

Практическая работа 5. Разработка программ с условными операторами

1. При регистрации на сайтах требуется вводить пароль дважды. Это сделано для безопасности, поскольку такой подход уменьшает возможность неверного ввода пароля. Напишите программу, которая сравнивает пароль и его подтверждение. Если они совпадают, то программа выводит: «Пароль принят», иначе: «Пароль не принят».
2. Напишите программу, которая определяет, является число четным или нечетным. На вход программе подается число, программа должна вывести «четное», если число четное и «нечетное» в противном случае.
3. Напишите программу, которая определяет, является ли год с данным номером високосным. Если год является високосным, то выведите «YES», иначе выведите «NO». Год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400.
4. Напишите программу, которая считывает три числа и подсчитывает сумму только положительных чисел. Если положительных нет, программа должна вывести 0.
5. Напишите программу, которая по возрасту пользователя определяет, разрешен ли ему доступ к интернет-ресурсу или нет. На вход программе подается возраст пользователя, если он больше 18, программа выводит «Доступ разрешен».
6. Напишите программу, которая принимает целое число x и определяет, принадлежит ли данное число промежуткам $(-30; -2]$ и $(7; 25]$.

Практическая работа 6. Вложенные и каскадные условные операторы

1. Напишите программу, которая принимает три положительных числа и определяет вид треугольника (равнобедренный, равносторонний, разносторонний), длины сторон которого равны введенным числам.
2. Даны три различных целых числа. Напишите программу, которая находит среднее по величине число. Решить задание используя циклы.
3. Известен вес боксера-любителя (целое число). Известно, что вес таков, что боксер может быть отнесён к одной из трех весовых категорий:
 - Легкий вес – до 60 кг (невключительно);
 - Первый полусредний вес – до 64 кг (невключительно);
 - Полусредний вес – до 69 кг (невключительно).
 Напишите программу, определяющую, в какой категории будет выступать данный боксер по введенному весу.
4. Дан порядковый номер месяца (1, 2, ..., 12). Напишите программу, которая выводит на экран количество дней в этом месяце. Принять, что год является невисокосным.
5. Каждую из задач решить двумя методами – с помощью каскадных и вложенных циклов.

Практическая работа 7. Разработка циклических программ

1. На вход программе подаются два целых числа a и b ($a \leq b$). Напишите программу, которая подсчитывает количество чисел в диапазоне от a до b включительно, куб которых оканчивается на 4 или 9.
2. На вход программе подается натуральное число n . Напишите программу, которая вычисляет сумму всех его делителей
3. Напишите программу, которая считывает натуральное число n и выводит первые n чисел последовательности Фибоначчи.
4. Дано натуральное число. Напишите программу, которая определяет, является ли последовательность его цифр при просмотре справа налево упорядоченной по неубыванию
5. На вход программе подается натуральное число n . Напишите программу, которая выводит числа от 1 до n включительно за исключением:
 - чисел от 5 до 9 включительно;
 - чисел от 17 до 37 включительно;
 - чисел от 78 до 87 включительно.
6. На вход программе подается число $n > 1$. Напишите программу, которая выводит его наименьший отличный от 1 делитель.
7. На вход программе подается последовательность целых чисел от 1 до 5, характеризующее оценку ученика, каждое число на отдельной строке. Концом последовательности является любое отрицательное число либо число, больше 5. Напишите программу, которая выводит количество пятерок.

Практическая работа 8. Вложенные циклы

1. Дано натуральное число n . Напишите программу, которая печатает численный треугольник в соответствии с примером:

```

1
22
333
4444
55555
...
```

2. На вход программе подается натуральное число n . Напишите программу, выводящую графическое изображение делимости чисел от 1 до n включительно. В каждой строке надо

напечатать очередное число и столько символов «+», сколько делителей у этого числа.
Пример вывода для введенной цифры 5:

```
1+
2++
3++
4+++
5++
```

3. На вход программе подается натуральное число n . Напишите программу, которая находит цифровой корень данного числа. Цифровой корень числа n получается следующим образом: если сложить все цифры этого числа, затем все цифры найденной суммы и повторять этот процесс до тех пор, пока в результате не будет получено однозначное число (цифра), которое и называется цифровым корнем изначального числа.
4. На вход программе подается два натуральных числа a и b ($a < b$). Напишите программу, которая находит все простые числа от a до b включительно. Натуральное число является простым, если оно отлично от 1 и делится без остатка только на 1 и на само на себя.

Практическая работа 9. Разработка программ со списками

1. На вход программе подается одно число n . Напишите программу, которая выводит список $[1, 2, 3, \dots, n]$.
2. Напишите программу, которая выводит сумму минимального и максимального элементов списка `numbers = [12.5, 3.1415, 2.718, 9.8, 1.414, 1.1618, 1.324]`.
3. Дополните приведенный код, используя операторы конкатенации (+) и умножения списка на число (*), так чтобы он вывел список:
`[1, 2, 3, 1, 2, 3, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 8, 9, 10, 11, 12, 13]`.

Исходный код:

```
numbers1 = [1, 2, 3]
numbers2 = [6]
numbers3 = [7, 8, 9, 10, 11, 12, 13]

print()
```

4. На вход программе подается натуральное число n , а затем n строк. Напишите программу, которая создает из указанных строк список и выводит его.
5. На вход программе подается натуральное число n . Напишите программу, которая создает список, состоящий из делителей введенного числа.
6. Дополните приведенный код так, чтобы элемент списка имеющий значение `Green` заменился на значение `Зеленый`, а элемент `Violet` на `Фиолетовый`. Далее необходимо вывести полученный список.

```
rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']

print()
```

Практическая работа 10. Разработка программ с использованием функций

1. Напишите функцию `draw_triangle()`, которая выводит звездный прямоугольный треугольник с катетами, равными 10 в соответствии с образцом:

```

*
**
***
****
*****
*****
*****
*****
*****
*****
*****

```

2. Напишите функцию `draw_triangle(fill, base)`, которая принимает два параметра:

- `fill` – символ заполнитель;
- `base` – величина основания равнобедренного треугольника;

а затем выводит его. Если ввести * и 9, то результат должен быть:

```

*
**
***
****
*****
*****
****
***
**
*

```

3. Напишите функцию `print_fio(name, surname, patronymic)`, которая принимает три параметра:

- `name` – имя человека;
- `surname` – фамилия человека;
- `patronymic` – отчество человека;

а затем выводит на печать ФИО человека.

4. Напишите функцию `convert_to_miles(km)`, которая принимает в качестве аргумента расстояние в километрах и возвращает расстояние в милях. Формула для преобразования: мили = километры * 0.6214.

5. Напишите функцию `is_valid_triangle(side1, side2, side3)`, которая принимает в качестве аргументов три натуральных числа, и возвращает значение `True` если существует невырожденный треугольник со сторонами `side1`, `side2`, `side3` и `False` в противном случае.

6. Напишите функцию `is_one_away(word1, word2)`, которая принимает в качестве аргументов два слова `word1` и `word2` и возвращает значение `True`, если слова имеют одинаковую длину и отличаются ровно в одном символе и `False` в противном случае.

7. Напишите функцию `get_circle(radius)`, которая принимает в качестве аргумента радиус окружности и возвращает два значения: длину окружности и площадь круга, ограниченного данной окружностью.

Практическая работа 11. Разработка программ с использованием вложенных списков

1. На вход программе подается число `n`. Напишите программу, которая создает и выводит построчно список, состоящий из `n` списков `[[1, 2, ..., n], [1, 2, ..., n], ..., [1, 2, ..., n]]`.

2. На вход программе подается строка текста, содержащая символы. Напишите программу, которая упаковывает последовательности одинаковых символов заданной строки в подсписки.
3. На вход программе подаются две строки: на одной – символы, на другой – число n . Из первой строки формируется список. Реализуйте функцию `chunked()`, которая принимает на вход список и число, задающее размер чанка (куска), а возвращает список из чанков (кусков) указанной длины.
4. На вход программе подаются два натуральных числа n и m , каждое на отдельной строке — количество строк и столбцов в матрице. Далее вводятся сами элементы матрицы — слова, каждое на отдельной строке; подряд идут элементы сначала первой строки, затем второй, и т.д.

Напишите программу, которая сначала считывает элементы матрицы один за другим, затем выводит их в виде матрицы.

5. Следом квадратной матрицы называется сумма элементов главной диагонали. Напишите программу, которая выводит след заданной квадратной матрицы.
6. Напишите программу, которая выводит количество элементов квадратной матрицы в каждой строке, больших среднего арифметического элементов данной строки. На вход программе подаётся натуральное число n — количество строк и столбцов в матрице, затем элементы матрицы (целые числа) построчно через пробел. Программа должна вывести n чисел — для каждой строки количество элементов матрицы, больших среднего арифметического элементов данной строки.
7. Напишите программу, которая проверяет симметричность квадратной матрицы относительно главной диагонали. На вход программе подаётся натуральное число n — количество строк и столбцов в матрице, затем элементы матрицы построчно через пробел. Программа должна вывести YES, если матрица симметрична относительно главной диагонали, и слово NO в противном случае.
8. Напишите программу для вычисления суммы двух матриц. На вход программе подаются два натуральных числа n и m — количество строк и столбцов в матрицах, затем элементы первой матрицы, затем пустая строка, далее следуют элементы второй матрицы. Программа должна вывести результирующую матрицу, разделяя элементы символом пробела.

Практическая работа 12. Обработка и методы работы с картежами и словарями

1. Дополните приведенный код так, чтобы он вывел произведение элементов кортежа `numbers`. `numbers = (2, 3, 5, 7, -11, 13, 17, 19, 23, 29, 31, -6, 41, 43, 47, 53, 59, 61, -96, 71, 1000, -1)`
2. Дополните приведенный код так, чтобы он вывел список, содержащий средние арифметические значения чисел каждого вложенного кортежа в заданном кортеже кортежей `numbers`. `numbers = ((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4), (90, 10))`
3. На вход программе подается натуральное число n , далее следует n строк с фамилией школьника и его оценкой на каждой из них. Программа должна вывести сначала все введенные строки с фамилиями и оценками учеников в том же порядке. Затем следует пустая строка, а затем выводятся строки с фамилиями и оценками хорошистов и отличников (в том же порядке).
4. Дополните приведенный код так, чтобы он вывел сумму минимального и максимального ключа в словаре `my_dict`. `my_dict = {1.12: 'aa', 67.9: 45, 3.11: 'ccc', 7.9: 'dd', 9.2: 'ee', 7.1: 'ff', 0.12: 'qq', 1.91: 'aa', 10.12: [1, 2, 3], 99.0: {9, 0, 1}}`
5. Напишите программу, которая по номеру курса выводит информацию о данном курсе. На вход программе подается одна строка – номер курса. Программа должна вывести номер курса, затем номер аудитории, имя преподавателя и время проведения курса в соответствии с примерами. Используйте словари.

Номер курса (ключ)	Номер аудитории (значение)	Преподаватель (значение)	Время (значение)
CS101	3004	Хайнс	8:00
CS102	4501	Альварадо	9:00
CS103	6755	Рич	10:00
NT110	1244	Берк	11:00
CM241	1411	Ли	13:00

Практическая работа 13. Использование дополнительных модулей в программе

1. Напишите программу, которая с помощью модуля `random` моделирует броски монеты. Программа принимает на вход количество попыток и выводит результаты бросков: Орел или Решка (каждое на отдельной строке).
2. Лотерейный билет содержит 7 чисел из диапазона от 1 до 49 (включительно). Напишите программу, которая с помощью модуля `random` генерирует 7 различных случайных чисел для лотерейного билета. Программа должна вывести числа в порядке возрастания на одной строке через один символ пробела.
3. Даны две дроби в формате `a/b`. Напишите программу, которая вычисляет и выводит их сумму, разность, произведение и частное.
4. На вход программе подается натуральное число `n`. Напишите программу, которая выводит в порядке возрастания все несократимые дроби, заключённые между 0 и 1, знаменатель которых не превосходит `n`. Возможно вам потребуется функция `gcd()`, которая позволяет находить наибольший общий делитель (НОД) двух чисел. Функция реализована в модуле `math`.
5. На вход программе подается `Decimal` число `d`. Напишите программу, которая вычисляет значение выражения:

$$e^d + \ln(d) + \lg(d) + \sqrt{d}$$

Тема 3. Основы объектно-ориентированного программирования

Практическая работа 15. Передача имен функций в качестве параметров

1. Написать программу, которая по запросу пользователя переводит число из десятичной системы в любую, из любой в десятичную, из двоичной в шестнадцатеричную и восьмеричную и обратно.
2. Составить математическую модель задачи
3. Для решения задачи использовать технологию передачи функций в качестве аргументов.

Практическая работа 16. Вложенные классы. Конструкторы Перегрузка

1. Создать класс, задаваемый тройкой координат, содержащий операции: сложение векторов, скалярное произведение векторов
2. Создать класс с закрытыми полями, где операции реализуются как методы класса.
3. Атрибуты инкапсулировать в отдельном классе. В базовом классе должно быть одно поле данных, представленное классом. Вложенный класс снабдить конструктором инициализации, перегрузить для него операции ввода/вывода, как внешние дружественные функции.
4. В базовом классе реализовать конструктор со списком инициализации и значениями по умолчанию и конструктор без аргументов

5. Указанные в задании операции реализуются посредством перегрузки подходящих операторов. В первую очередь реализуются как методы класса, в противном случае как внешние дружественные функции.

6. Интерфейс класса и реализация должны быть вынесены в отдельные файлы, за пределы файла с главной функцией.

Практическая работа 17. Множественное наследование. Полиморфизм

1. Получить у преподавателя индивидуальное задание из тройки классов: Базовый – Первый уровень наследования – Второй уровень наследования.

2 Спроектировать:

- архитектуру классов;
- компоненты классов;

Соблюдать следующие требования:

- Базовый класс должен содержать минимум два атрибута и минимум два метода, один из которых чисто виртуальный.
- Открытый метод базового класса должен перекрываться в классах наследниках.
- Наследники имеют свои индивидуальные атрибуты и методы.
- Второй уровень наследования – множественный.
- Использовать различные модификаторы доступа и наследования.
- Снабдить классы конструкторами и деструкторами (виртуальными, где необходимо)
- Интерфейс класса и реализация должны быть разделены и вынесены в отдельные файлы, за пределы файла с главной функцией.

Варианты индивидуальных заданий к практической № 17:

1. Персонаж игры – Архетип – Комбинации
2. Транспорт – Среда передвижения – Комбинации
3. Летательный аппарат – Движитель – Комбинации
4. Сухопутный транспорт – Движитель – Комбинации
5. Морской транспорт – Движитель – Комбинации
6. Космический аппарат – Движитель – Комбинации
7. Источник питания – Способ преобразования энергии – Приборы использующий несколько источников
8. Человек – Профессия – Комбинации
9. Книга – Содержимое страницы – Комбинации
10. Средство связи – Вид передаваемой информации – Комбинации
11. Строение – Тип помещения – Комбинации
12. Кухонная утварь – Способ готовки – Комбинации
13. Кухонная техника – Функции – Комбинации
14. Материал – Физические свойства – Комбинации
15. Организм – Среда обитания – Комбинации

Примеры тестовых заданий

Тема 2. Основы программирования

Вопрос 1. Какой тип данных в Python представляет целые числа?

- а) complex
- б) float
- в) int
- г) bool.

Вопрос 2. Какие из представленных литералов чисел относятся к типу float?

- а) 1.7+4.3j
- б) 5.0
- в) 88
- г) -.4

Вопрос 3. Какой тип данных в Python представляет кортежи?

- а) dict
- б) list
- в) tuple
- г) set.

Вопрос 4. Как в Python создать пустой кортеж?

- а) tpl = ()
- б) tpl = {}
- в) tpl = tuple()
- г) tpl = tuple(0).

Вопрос 5. Выберите верные утверждения о кортежах.

- а) неизменяемые
- б) изменяемые
- в) неупорядоченные
- г) упорядоченные.

Вопрос 6. Что будет выведено на экран в результате выполнения кода условия?

```
tpl = (1, 2, ('-3-4-5-6', '-7-8-9-10'))
print(tpl[2][1][:-1::-2])
```

- а) 0----
- б) 76543
- в) 10987
- г) ----.

Вопрос 7. Какое значение будет иметь выражение True and False?

- а) True
- б) False
- в) None
- г) ошибка.

Вопрос 8. Какие из частей условной инструкции не являются обязательными?

- а) if
- б) elif
- в) else
- г) все части необязательны.

Вопрос 9. Что будет выведено на экран в результате выполнения кода условия?

```
очки = 73

if очки >= 90:
    разряд = "1-й разряд"
elif очки >= 80:
    разряд = "2-й разряд"
elif очки >= 70:
    разряд = "3-й разряд"
else:
    разряд = "не присваивается"
```

```
print(разряд)
```

- а) 1-й разряд
- б) 2-й разряд
- в) 3-й разряд
- г) не присваивается

Вопрос 10. Какой из предлагаемых операторов можно использовать для инвертирования условия в инструкции if?

- а) and
- б) or
- в) not
- г) is

Вопрос 11. Какие из представленных операторов представляют циклы в Питоне?

- а) if
- б) while
- в) switch
- г) for.

Вопрос 12. Что будет выведено на экран в результате выполнения кода условия?

```
for i in (3):
    print(i, end=' ')
```

- а) 3
- б) (3)
- в) 1 2 3
- г) ошибка.

Вопрос 13. Какой из следующих операторов предназначен для прерывания цикла?

- а) pass
- б) continue
- в) break
- г) else.

Вопрос 14. Что будет выведено на экран в результате выполнения кода условия?

```
k = 0
while True:
    if k > 5: break
    if k == 3:
        k += 1
        continue
    print(k, end=' ')
    k += 1
```

- а) 1 2 3
- б) 1 2 3 4
- в) 1 2 3 4 5
- г) ошибка.

Вопрос 15. Какой из следующих операторов можно использовать для выполнения блока кода после завершения цикла?

- а) pass
- б) continue
- в) break

г) else

Вопрос 16. Что будет выведено на экран в результате выполнения кода условия?

```
k = 1
while True:
    for n in range(2, 0, -1):
        print(n, end=' ')
    k += 1
    if k == 3: break
```

- а) 2 1 2 1
- б) 2 1 2 1 2 1
- в) ничего
- г) ошибка.

Вопрос 17. Какая инструкция используется в Python для определения функций?

- а) df
- б) def
- в) dfn
- г) define.

Вопрос 18. Что возвращает функция, в которой отсутствует инструкция return?

- а) объект функции
- б) None
- в) False
- г) ничего.

Вопрос 19. Что будет выведено на экран в результате выполнения кода условия?

```
def my_func():
    print(True)
my_func
```

- а) True
- б) None
- в) ничего
- г) ошибка.

Вопрос 20. Какой из предлагаемых вариантов является допустимым способом передачи аргументов функции?

- а) по порядку
- б) по ключевому слову
- в) по умолчанию
- г) все вышеуказанные.

Вопрос 21. Какие функции в Питоне называют рекурсивными?

- а) функция, которая вызывает саму себя внутри своего тела
- б) функция, которая вызывает другую функцию внутри своего тела
- в) функция, которая одновременно не имеет параметров и возвращаемого значения
- г) рекурсивные функции в Пайтоне запрещены.

Вопрос 22. Какие из утверждений о словарях верны?

- а) изменяемые
- б) неизменяемые
- в) доступ к элементам по индексу
- г) доступ к элементам по ключу.

Вопрос 23. Какие из представленных литералов относятся к словарям?

- а) {'красный', 'синий'}
- б) {1: 'Январь', 2: 'Февраль'}

- в) ({'name': 'Оля'}, {'age': 18})
- г) {'list': [1, 2, 3]}.

Вопрос 24. Что будет выведено на экран в результате выполнения кода условия?

```
d = {1: 'a', '2': ['b', {'c': 3}], (5, '6'): ('f', 'g')}
print(len(d))
```

- а) 2
- б) 3
- в) 4
- г) ошибка.

Вопрос 25. Как получить значение из словаря d по ключу key?

- а) d.value(key)
- б) d[key]
- в) d.get(key)
- г) d.setdefault(key).

Тема 3. Основы объектно-ориентированного программирования

Вопрос 1. Укажите принципы объектно-ориентированного программирования:

- а) Инкапсуляция, наследование, полиморфизм
- б) абстракция, формальность
- с) достоверность, полнота, ценность
- б) разделение на модули, стандартизация, гибкость

Вопрос 2. Инкапсуляция – это...

- а) Такое свойство, при котором объекты содержат описание атрибутов и действий одновременно
- б) Набор правил и спецификаций, определяющих, каким образом модули могут взаимодействовать друг с другом.
- с) Такое свойство объектов, при котором действие с одинаковыми именами вызывает различное поведение для различных объектов

Вопрос 3. Укажите параметр, характеризующий объект, обладающий свойством инкапсуляции:

- а) Набор атрибутов
- б) Набор действий для описания своего поведения
- с) Уникальное имя
- д) Все выше перечисленные

Вопрос 4. Полиморфизм – это...

- а) Такое свойство, при котором объекты содержат описание атрибутов и действий одновременно
- б) Набор правил и спецификаций, определяющих, каким образом модули могут взаимодействовать друг с другом.
- с) Такое свойство объектов, при котором действие с одинаковыми именами вызывает различное поведение для различных объектов

Вопрос 5. Конструктор класса задается методом с именем:

- а) __new__
- б) __init__
- с) __construct__
- д) new
- е) init
- ф) имя конструктора совпадает с именем класса

Вопрос 6. Атрибуты объекта - это...

- a) Данные, характеризующие состояние объекта
- b) Представители класса, построенные по хранящемуся в классе описанию.
- c) Функции (или преобразование), которые можно применять к объектам данного класса.

Вопрос 7. Укажите верное утверждение.

- a) Полиморфная операция не может применяться к объектам разных классов
- b) Наследование заключается в том, что класс-наследник приобретает свойства и методы родительского класса, но не добавляет к ним собственные
- c) Каждый объект задается именем, свойствами и методами
- d) Наличие в классе нескольких конструкторов называется аддитивностью

Вопрос 8. Совокупность объектов, характеризующаяся общностью методов и свойств – это...

- a) операция
- b) класс
- c) конструктор
- d) модуль

Вопрос 9. Изменение состояния объекта в ответ на какое-либо действие – это...

- a) событие
- b) наследование
- c) итерация
- d) редактирование

Вопрос 10. Для того, чтобы функция, определенная для типа А работала с типом В необходимо:

- a) чтобы тип В был наследником типа А
- b) для типа В были определены все методы класса А
- c) для типа В были определены все методы класса А, применяющиеся в функции
- d) в языке Python функции работают только с теми типами, для которых определены

Вопрос 11. Чем отличаются в языке Python обычные модули от модулей расширения?

- a) тем, что модули расширения могут состоять из нескольких файлов
- b) способом подключения к главному модулю
- c) языком программирования, на котором они написаны
- d) ничем не отличаются

Вопрос 12. При импортировании модуля если в пространстве имен уже существует имя импортированного модуля, то:

- a) модуль импортируется и заменяет своим именем уже существующее
- b) модуль импортируется только в случае, если определенное имя не является именем модуля
- c) будет выдана ошибка
- d) ошибки выдано не будет, но модуль не импортируется

Вопрос 13. В языке Python доступ через `instance.attribute` разрешен к любым атрибутам кроме:

- a) атрибутов с идентификаторами `private`
- b) атрибутов, имена которых начинаются и кончаются на подчеркивание
- c) атрибутов, имена которых начинаются и кончаются на двойное подчеркивание
- d) атрибутов, имена которых начинаются на двойное подчеркивание и кончаются не на подчеркивание

Вопрос 14. При наследовании в языке Python порядок разрешения методов для "новых" классов (т.е. классов, наследников класса `object`) определяется следующим образом:

- a) рассматривается первый суперкласс и далее его суперкласс, если метод не найден, то рассматривается второй суперкласс
- b) рассматривается первый суперкласс, если метод не найден, то рассматривается второй суперкласс и т.д., если во всех суперклассах метод не найден, то рассматривается суперклассы первого суперкласса и т.д.

- с) рассматривается последний суперкласс и далее его суперкласс, если метод не найден, то рассматривается предпоследний суперкласс
- д) рассматривается последний суперкласс, если метод не найден, то рассматривается предпоследний суперкласс и т.д., если во всех суперклассах метод не найден, то рассматривается суперклассы последнего суперкласса и т.д.

Вопрос 15. Если в классе определены два метода с одинаковыми именами и разными списками параметров, то:

- а) при выполнении скрипта будет сгенерирована ошибка
- б) будет сгенерировано предупреждение, второе определение заменит первое
- с) не будет сгенерировано ни предупреждения, ни ошибки; второе определение заменит первое
- д) не будет сгенерировано ни предупреждения, ни ошибки; вызов того или иного метода будет зависеть от типа и количества указанных при вызове параметров
- е) будет сгенерировано предупреждение; вызов того или иного метода будет зависеть от типа и количества указанных при вызове параметров

1. »

- 2. Разработка модуля ИС «Планирование бюджета»
- 3. Разработка модуля ИС «Электронный переводчик»
- 4. Разработка модуля ИС «Управление портфелем заказов»

Темы можно модифицировать, можно предлагать свои.

3.3 Оценочные средства промежуточной аттестации обучающихся

Вопросы для подготовки к устному экзамену:

1. Алгоритм. Определение, свойства, способы представления алгоритма.
2. Запись алгоритма блок-схемами. Основные элементы блок-схем.
3. Этапы решения задачи на ЭВМ.
4. Базовые алгоритмические структуры. Линейная структура. Оператор присваивания. Примеры линейной структуры.
5. Базовые алгоритмические структуры. Структура ветвления. Примеры структуры ветвления.
6. Базовые алгоритмические структуры. Циклическая структура. Типы циклов. Цикл с параметром. Примеры
7. Базовые алгоритмические структуры. Циклическая структура. Типы циклов. Цикл с постусловием. Примеры
8. Базовые алгоритмические структуры. Циклическая структура. Типы циклов. Цикл с предусловием. Примеры
9. Базовые алгоритмические структуры. Циклическая структура. Типы циклов. Вложенные циклы. Примеры
10. Основные типы данных.
11. Булевый тип данных. Операции с переменными этого типа.
12. Синтаксис, семантика, алфавит и лексика программирования. Приведите пример.
13. Операторы break и continue. Назначение и примеры использования
14. Списки. Создание и вывод, методы работы со списками.
15. Сортировка списков. Задача сортировки, классификация алгоритмов.
16. Сортировка списков. Пузырьковая сортировка.
17. Сортировка списков. Сортировка выбором
18. Сортировка списков. Сортировка простыми вставками
19. Функции. Виды и преимущества использования функций.
20. Локальные и глобальные переменные.
21. Матрицы. Методы работы с матрицами

22. Кorteжи и множества
23. Базовые понятия ООП: объект, метод, класс, интерфейс
24. Основные принципы ООП и их характеристика
25. Классы объектов. Компоненты и их свойства
26. Иерархия классов
27. Событийно-управляемая модель программирования
28. Компонентно-ориентируемый подход
29. Принципы разработки программного обеспечения
30. Жизненный цикл ПО. Этапы разработки программного обеспечения

4 Методические указания для обучающихся по освоению дисциплины

Приступая к изучению дисциплины «Основы алгоритмизации и программирования», студент должен ознакомиться с содержанием данной «Рабочей учебной программы дисциплины» с тем, чтобы иметь четкое представление о своей работе. Изучение дисциплины осуществляется на основе выданных студенту преподавателем рекомендаций по выполнению всех заданий, предусмотренных учебным планом и программой.

В первую очередь необходимо уяснить цель и задачи изучаемой дисциплины, оценить объем материала, познакомиться с предложенной и подобрать основную и дополнительную литературу, выявить наиболее важные проблемы, стоящие по вопросам изучаемой дисциплины.

Выполнение заданий осуществляется в соответствии с учебным планом и программой. Они должны выполняться в соответствии с методическими рекомендациями, выданными преподавателем, и представлены в установленные преподавателем сроки.

Работая с учебниками и учебными пособиями, целесообразно законспектировать тот материал, который не сообщался студентам на лекциях.

На занятиях лекционного и практического характера студентам для работы требуется тетрадь для записи лекций и заданий.

Для успешного овладения программой дисциплины необходимо выполнять следующие требования:

- посещать все лекционные и практические занятия;
- все рассматриваемые на лекциях и практических занятиях темы и вопросы обязательно фиксировать в тетради;
- в случае пропуска занятий по каким-либо причинам необходимо обязательно самостоятельно изучать соответствующий материал в Moodle, фиксируя записи в тетради, а также выполнять практические задания.

Подготовка к зачету с оценкой осуществляется по представленным в списке основной и дополнительной литературе, а также частично по нормативным документам. В учебниках и учебных пособиях содержатся одноименные параграфы, что позволит успешно

подготовиться к зачету с оценкой. Рекомендуемые литература и интернет-ресурсы будут полезны при выполнении практических заданий, при чтении кода программ и для подготовки к тестированиям. В рекомендуемых интернет-ресурсах также можно найти ссылки на научные журналы по информационным технологиям и на форумы профессиональных сообществ программистов.

Методические рекомендации по составлению конспекта

Конспект — сложный способ изложения содержания книги или статьи в логической последовательности. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта. Выделите главное, составьте план, представляющий собой перечень заголовков, подзаголовков, вопросов, последовательно раскрываемых затем в конспекте.

Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

При оформлении конспекта необходимо стремиться к емкости каждого предложения.

Методические рекомендации по составлению опорного конспекта

Опорный конспект — вид внеаудиторной самостоятельной работы студента по созданию краткой информационной структуры, обобщающей и отражающей суть материала лекции, темы учебника.

Опорный конспект — это наилучшая форма подготовки к ответу на вопросы.

Основная цель опорного конспекта — облегчить запоминание. Этапы составления опорного конспекта:

1. Изучить материалы темы, выбрать главное и второстепенное;
2. Установить логическую связь между элементами темы;
3. Представить характеристику элементов в краткой форме;
4. Выбрать опорные сигналы для акцентирования главной информации и отобразить в структуре работы.

Методические рекомендации по прохождению тестирования

Тестирование — это исследовательский метод, который позволяет выявить уровень знаний, умений и навыков, способностей, а также их соответствие определенным нормам усвоения, путем выполнения испытуемым ряда специальных заданий.

Следует понимать, что тестовые задания могут быть представлены в различных формах:

— задания закрытой формы, в которых обучающийся выбирает один или несколько правильных ответов из заданного набора:

— задания на дополнение (открытые задания) требующие самостоятельного получения ответов:

— задания на установления соответствия (с множественным выбором), выполнение которых связано с выявлением соответствия между элементами нескольких множеств:

— задания на установление правильной последовательности, в которых от учащегося требует указать порядок действий или процессов и другие. Этапы подготовки к тестированию:

1. Внимательно прочитайте материал по конспекту, составленному на учебном занятии. Прочитайте тот же материал по учебнику, учебному пособию.

2. Постарайтесь разобраться с непонятным, в частности новыми терминами и конструкциями.

3. Ответьте на контрольные вопросы для самопроверки, имеющиеся в учебнике, конспекте и т. д.

4. Кратко перескажите содержание изученного материала «своими словами».

5. Выучите определения основных понятий, условные обозначения, формулы и конструкции.

Подготовка к практическим занятиям

В ходе подготовки к практическим занятиям необходимо изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях, ознакомиться с программным обеспечением. Следует дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

Заканчивать подготовку следует закреплением материала с использованием соответствующих программных продуктов.

Все практические задания, предусмотренные рабочей программой, представлены в фонде оценочных средств по дисциплине.

Критерии оценивания выполненных практических работ:

— правильность выполнения работы (отсутствие фактических, логических и других ошибок);

— полнота выполнения работы;

— своевременность выполнения;

— правильность оформления отчета.

За задания, выполненные позже установленного срока или с нарушениями требований к оформлению, оценка на балл снижается.

Порядок организации самостоятельной работы студентов

Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Целью самостоятельной работы студентов является: овладение практическими знаниями, профессиональными умениями и навыками деятельности по специальности, опытом творческой, исследовательской деятельности.

Самостоятельная работа студентов предполагает:

- самостоятельный поиск ответов и необходимой информации в рамках изучаемых тем;
- выполнение заданий для самостоятельной работы, в том числе тестов;
- изучение теоретического и лекционного материала, а также основной и дополнительной литературы при подготовке к практическим занятиям.

Самостоятельная работа студентов является обязательным элементом подготовки специалиста среднего звена.

5 Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»

Основная литература

1. Абдрахманов, М. И. Основы языка программирования Python : учебное пособие для СПО / М. И. Абдрахманов. — Москва : Ай Пи Ар Медиа, 2023. — 142 с. — ISBN 978-5-4497-2310-9. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROОбразование : [сайт]. — URL: <https://profspo.ru/books/132567>
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1927269>
3. Гуриков, С. Р. Основы алгоритмизации и программирования на Visual C++ : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 515 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1039154. - ISBN 978-5-16-015500-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1039154>
4. Дорогов, В. Г. Основы программирования на языке C : учебное пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0809-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2010597>

5. Дорохова, Т. Ю. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Ю. Дорохова, И. Е. Ильина. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 139 с. — ISBN 978-5-4488-1531-7, 978-5-4497-1718-4. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROобразование : [сайт]. — URL: <https://profspo.ru/books/122426>

6. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1735805>

Дополнительная литература

1. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2020. — 137 с— URL: <https://urait.ru/bcode/454452>

2. Семакин И.Г. Основы алгоритмизации и программирования. Практикум: учебное издание / Семакин И.Г., Шестаков А. П. - Москва : Академия, 2023. - 144 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow»

3. Семакин И.Г. Основы алгоритмизации и программирования: учебное издание / Семакин И.Г., Шестаков А. П. - Москва : Академия, 2024. - 304 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow».

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

Для реализации дисциплины необходимы:

Лаборатория программирования и баз данных, включающая:

- компьютерный класс (15-20 рабочих мест) с современными ПК, объединенными в локальную сеть с выходом в Интернет;
- проектор и экран;
- маркерная или меловая доска;
- информационно-коммуникационные технологии;
- программное обеспечение.

Информационно-коммуникационные технологии

Локальная вычислительная сеть с организованным доступом к электронным образовательным ресурсам.

Система видеоконференцсвязи (Mts Link) для проведения дистанционных консультаций.

Виртуальная образовательная среда на базе LMS (Moodle) для размещения учебных материалов и проведения тестирования.

Система облачного хранения (Яндекс Диск) и системы для коллективной работы над проектами (Яндекс Документы, Mts Link.Доски).

Программное обеспечение

Windows 10/11, Linux (дистрибутивы Debian, Ubuntu, Astra или Alt), MS Visual Studio, MS Visual Studio Code, SQLite, PostgreSQL, pgAdmin, DBeaver, Python IDE, 1С:Предприятие, Eclipse IDE, Apache NetBeans, Spacemacs, AndroidStudio, Draw.io, StarUML 5, Inkscape, LibreOffice, Oracle VM VirtualBox, Notepad++.

Электронно-библиотечные системы (ЭБС)

1. Электронно-библиотечная система «IPR SMART» (СПО) — URL: <https://www.iprbookshop.ru/>;
2. Электронно-библиотечная система «Лань» — URL: <https://e.lanbook.com/>;
3. Электронно-библиотечная система «iBooks» — URL: <https://ibooks.ru/>;
4. Электронно-библиотечная система «ZNANIUM» — URL: <https://znanium.ru/>.