

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шутов Олег Леонтьевич

Должность: Директор

Дата подписания: 06.06.2026 11:53:57

Уникальный программный ключ:

2ee6ded937fc2877009a3b03e0f0a7f33d8083d5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ПРОФЕССИОНАЛЬНАЯ  
ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»  
(АНПО «КУБАНСКИЙ ИПО»)**

**ОТДЕЛЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

## **РАБОЧАЯ ПРОГРАММА**

учебной дисциплины

**ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

по специальности

**09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ  
ОБЕСПЕЧЕНИЕМ**

**направленность программы: Разработка информационных систем**

Краснодар, 2026

**СОГЛАСОВАНО**

Зам. директора по КОД и МР

\_\_\_\_\_/ Т.В. Першакова  
28.05.2026 г.**УТВЕРЖДАЮ**

Директор АНПОО «Кубанский ИПО»

\_\_\_\_\_/ О.Л. Шутов  
Приказ №38-О от 28.05.2026 г.**ОДОБРЕНО**

Педагогическим советом

Протокол №6 от 28.05.2026 г

**РАССМОТРЕНО**

на заседании УМО

«Информационные системы и  
программирование»

Протокол № 5 от 15.05.2026г.

Председатель \_\_\_\_\_ / С.А. Пясецкий

Рабочая программа учебной дисциплины ОП.06 Основы алгоритмизации и программирования предназначена для реализации образовательной программы подготовки специалистов среднего звена.

Разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением (Приказ Министерства Просвещения Российской Федерации от 24 февраля 2025 г. № 138, зарегистрированного Министерством Юстиции России 31 марта 2025 г. № 81696) с учетом примерной образовательной программы, разработанной Федеральным учебно-методическим объединением в системе среднего профессионального образования по укрупненным группам профессий, специальностей 09.00.00 Информатика и вычислительная техника, с учетом профессиональных стандартов: «Программист» (Приказ Министерства труда и социальной защиты РФ от 20 июля 2022 г. № 424н, зарегистрирован Министерством юстиции Российской Федерации от 22 августа 2022г. №69720); «Специалист по информационным системам» (Приказ Министерства труда и социальной защиты РФ от 13 июля 2023 г. № 586н, зарегистрирован Министерством юстиции Российской Федерации от 16 августа 2023 г № 74817) и компетенции «Программные решения для бизнеса».

**Организация-разработчик:** АНПОО «Кубанский ИПО»

**Разработчики:**

Пясецкий С.А., преподаватель АНПОО «Кубанский ИПО»

Бондаренко А.Ю., преподаватель АНПОО «Кубанский ИПО»

**Рецензенты:**

1. Варкентин В.Ф. – преподаватель, АНПОО «Кубанский ИПО»

Квалификация по диплому: преподаватель информатики

2. Маслиев Р.О, генеральный директор ООО «Старт Эксперт»

## СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	14
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ ..	16

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

## 1.1 Место дисциплины в структуре основной образовательной программы:

Учебная дисциплина ОП.06 Основы алгоритмизации и программирования является обязательной частью общепрофессионального цикла образовательной программы в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

Особое значение дисциплина имеет при формировании и развитии общих и профессиональных компетенций:

ОК.01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК.02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК.04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК.09. Пользоваться профессиональной документацией на государственном и иностранном языке;

ПК 2.1. Проектировать модули программного обеспечения;

ПК 2.2. Разрабатывать модули программного обеспечения;

ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения;

## 1.2 Цель и планируемые результаты освоения дисциплины:

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания.

Код ПК, ОК	Умения	Знания
ПК.2.1., ПК.2.2., ПК.2.3., ОК.01, ОК.02, ОК.04, ОК.09	<ul style="list-style-type: none"><li>– распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части;</li><li>– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации;</li><li>– взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности;</li><li>– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы;</li><li>– разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;</li><li>– анализировать требования и определять функциональность модуля;</li><li>– создавать интерфейсы для взаимодействия с другими модулями и системами;</li><li>– обеспечивать безопасность, производительность и масштабируемость при разработке модулей;</li><li>– оптимизировать проектируемые модули для повышения их эффективности и качества;</li><li>– определять структуру и интерфейсы модулей;</li></ul>	<ul style="list-style-type: none"><li>– актуальный профессиональный и социальный контекст, в котором приходится работать и жить;</li><li>– номенклатура информационных источников, применяемых в профессиональной деятельности;</li><li>– язык программирования, основные конструкции, синтаксис;</li><li>– структуры данных;</li><li>– принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP;</li><li>– работа с инструментальным программным обеспечением;</li><li>– многопоточность в программных модулях;</li><li>– основные принципы проектирования модулей программного обеспечения;</li><li>– языки программирования и технологии для реализации модулей;</li><li>– принципы создания интерфейсов для взаимодействия с другими модулями и системами;</li><li>– принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей;</li><li>– методы и подходы к интеграции модулей и компонентов;</li></ul>

<ul style="list-style-type: none"> <li>– анализировать требования к модулю и определять его функциональность;</li> <li>– создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля;</li> <li>– проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;</li> <li>– учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля;</li> <li>– интегрировать модули и компоненты, обеспечивая их взаимодействие;</li> <li>– работать с различными форматами данных и протоколами передачи данных;</li> <li>– <i>разрабатывать консольные приложения на Python с меню и обработкой ввода;</i></li> <li>– <i>использовать списки, словари, множества и строки для обработки данных;</i></li> <li>– <i>применять ветвления, циклы, функции и рекурсию при решении алгоритмических задач;</i></li> <li>– <i>выполнять чтение и запись текстовых файлов и JSON;</i></li> <li>– <i>создавать классы и объекты с инкапсуляцией, наследованием и полиморфизмом;</i></li> <li>– <i>использовать Git (коммиты, ветки, слияние) для управления проектами;</i></li> <li>– <i>отлаживать код с помощью pdb и логирования;</i></li> <li>– <i>обрабатывать исключения (try-except);</i></li> <li>– <i>применять lambda, map, filter и списковые включения;</i></li> <li>– <i>разрабатывать законченное консольное приложение с тестами (unittest) и хранением данных в JSON.</i></li> </ul>	<ul style="list-style-type: none"> <li>– принципы безопасности при интеграции модулей и компонентов.</li> <li>– <i>свойства алгоритмов, способы записи, основы O-нотации;</i></li> <li>– <i>особенности Python: динамическая типизация, изменяемые/неизменяемые типы;</i></li> <li>– <i>синтаксис Python: переменные, операторы, ветвления, циклы, ввод/вывод;</i></li> <li>– <i>структурированные типы: list, dict, set, tuple;</i></li> <li>– <i>работа с файлами (open, with) и JSON (json.load, json.dump);</i></li> <li>– <i>модульное программирование (import, if name == «main»);</i></li> <li>– <i>рекурсия (базовый случай, шаг рекурсии);</i></li> <li>– <i>основы ООП: классы, объекты, init, наследование, полиморфизм, инкапсуляция (property);</i></li> <li>– <i>основы Git (init, add, commit, branch, merge);</i></li> <li>– <i>отладка (pdb) и логирование (logging);</i></li> <li>– <i>функциональные возможности: lambda, map, filter, list comprehensions;</i></li> <li>– <i>исключения (try-except-else-finally) и контекстные менеджеры (with);</i></li> <li>– <i>жизненный цикл приложения, основы тестирования (unittest), валидация ввода.</i></li> </ul>
---	---

## 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1 Объем учебной дисциплины и виды учебной работы

<b>Вид учебной работы</b>	<b>Объем в часах</b>
<b>ОБЪЕМ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ДИСЦИПЛИНЫ</b>	<b>144</b>
в том числе вариативная часть	72
- теоретическое обучение	<b>54</b>
- практические занятия	<b>72</b>
в т.ч. в форме практической подготовки	72
- промежуточная аттестация	<b>18</b>
в том числе:	
консультации	12
экзамен	6

## Тематический план учебной дисциплины

Наименование разделов и тем	Количество аудиторных часов				
	Всего	в т.ч. в форме практической подготовки	самост. работа	теоретич. обучение	практич. занятия
<b>Раздел 1. Введение в программирование</b>	<b>46</b>	<b>26</b>	-	<b>20</b>	<b>26</b>
Тема 1.1 Основы алгоритмизации, языки и системы программирования. Основные элементы языка. Типы данных. Основы структурного программирования	46	26	-	20	26
<b>Раздел 2. Технологии программирования</b>	<b>44</b>	<b>26</b>	-	<b>18</b>	<b>26</b>
Тема 2.1 Модульное программирование	20	6	-	14	6
<b><i>Итого за семестр 3</i></b>	<b>66</b>	<b>32</b>	-	<b>34</b>	<b>32</b>
Тема 2.1 Модульное программирование	24	20	-	4	20
<b>Раздел 3. Разработка приложений</b>	<b>36</b>	<b>20</b>	-	<b>16</b>	<b>20</b>
Тема 3.1 Этапы разработки приложений	36	20	-	16	20
<b>Консультации</b>	<b>12</b>	-	-	-	-
<b>Экзамен</b>	<b>6</b>	-	-	-	-
<b><i>Итого за семестр 4</i></b>	<b>78</b>	<b>40</b>	-	<b>20</b>	<b>40</b>
<b>ВСЕГО</b>	<b>144</b>	<b>72</b>	-	<b>54</b>	<b>72</b>

## 2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию которых способствует элемент программы	
1	2	3	4	
<b>СЕМЕСТР 3</b>				
<b>Раздел 1. Введение в программирование</b>		<b>46</b>		
<b>Тема 1.1</b> Основы алгоритмизации, языки и системы программирования. Основные элементы языка. Типы данных. Основы структурного программирования.	<b>Содержание учебного материала</b>		ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09	
	1	<b>Алгоритмы и их свойства</b> Понятие алгоритма, свойства (массовость, конечность, детерминированность и др.). Способы записи: словесный, графический (блок-схемы), псевдокод, программный. Сложность алгоритмов (O-нотация).		2
	2	<b>Языки программирования и среда разработки</b> Эволюция языков программирования. Классификация: машинные, ассемблеры, высокого уровня. Компиляторы и интерпретаторы. Среда программирования (IDE). Жизненный цикл программы. Программный продукт и его характеристики. <i>Роль Python в разработке информационных систем: обработка данных, автоматизация, прототипирование, бэкенд-логика (базовое представление). Инструменты разработки на Python: интерактивная среда (IDLE), редакторы (VS Code, PyCharm Community), Jupyter Notebook для анализа данных. Жизненный цикл программы. Программный продукт и его характеристики.</i>		2
	3	<b>Типы данных и переменные</b> Основные элементы языка. Простые типы данных (целые, вещественные, логические, символьные). Производные типы. Структурированные типы. Объявление переменных. Константы. <i>Базовый синтаксис Python. Простые типы данных в Python: int, float, bool, str. Производные и структурированные типы: list, tuple, dict, set. Особенности Python: динамическая типизация, приведение типов, функции type(), isinstance(). Объявление переменных (без ключевых слов). Ввод/вывод в консольном приложении: input(), print().</i>		2
	4	<b>Операторы и ввод/вывод</b> Арифметические, логические, реляционные операторы. Ввод/вывод данных. Обработка исключений. Операторы ветвления (if, if-else, switch). <i>Особенности: //, %, *, and/or/not, сравнение цепочек (a &lt; b &lt; c). Обработка исключений: try-except-else-finally. Перехват конкретных типов ошибок (ValueError, TypeError). Операторы ветвления: if, elif, else. Тернарный оператор x if условие else y.</i>		2
5	<b>Циклические конструкции</b>	2		

	Циклы for, while, do-while. Вложенные циклы. Управление циклом (break, continue). Бесконечные циклы. <i>Условные операторы: if, elif, else. Циклы: for, while, вложенные циклы, операторы break и continue.</i>	
6	<b>Массивы</b> Одномерные массивы: объявление, инициализация, обход. Двумерные массивы: представление в памяти, обход по строкам и столбцам. Базовые алгоритмы обработки.	2
7	<b>Строки и коллекции</b> Строковые типы, основные операции. Коллекции: списки (List), словари (Map/Dictionary), множества (Set). Сравнение массивов и коллекций. <i>Строки. Индексация. Срезы.</i>	2
8	<b>Файловая система и работа с файлами</b> Виды файлов (текстовые, бинарные). Доступ к файлам. Режимы открытия. Чтение и запись. Работа с каталогами. <i>Числовые данные, логические значения. Дата и время. Работа с файлами.</i>	2
9	<b>Введение в обработку данных из внешних источников.</b> <i>Чтение структурированных данных из текстового файла. Работа с аргументами командной строки (sys.argv) как простейший интерфейс информационной системы.</i>	2
10	<b>Форматы обмена данными.</b> <i>Структура JSON, типы данных в JSON. Библиотека json в Python: json.loads, json.dumps. Чтение JSON из файла. XML – базовый синтаксис (элементы, атрибуты).</i>	2
<b>в том числе, практических занятий</b>		<b>26</b>
	<b>*ПЗ №1.</b> Создание линейных программ. <i>Ввод и вывод данных в консоли (input, print). Вычисление выражений.</i>	2
	<b>*ПЗ №2.</b> Составление программ разветвляющейся структуры. <i>Условный оператор if-elif-else.</i>	2
	<b>*ПЗ №3.</b> Составление циклических программ. <i>Циклы for и while. Организация меню информационной системы.</i>	2
	<b>*ПЗ №4.</b> Работа с одномерными массивами. <i>Создание, добавление, удаление элементов. Линейный поиск. Подсчёт статистик (сумма, среднее, минимум, максимум).</i>	2
	<b>*ПЗ №5.</b> Работа с многомерными массивами. <i>Создание и вывод таблицы. Обработка строк и столбцов.</i>	2
	<b>*ПЗ №6.</b> Обработка массивов. <i>Обработка коллекций. Фильтрация списка по условию.</i>	2
	<b>*ПЗ №7.</b> Сортировка массивов. <i>Встроенная функция sorted(), метод list.sort(). Обратная сортировка (reverse=True). Сортировка по ключу (lambda).</i>	2
	<b>*ПЗ №8.</b> Работа с символами и строками. <i>Обработка строк. Методы строк (split, join, strip, find, replace).</i>	2
	<b>*ПЗ №9.</b> Использование коллекций. <i>Словарь как аналог простой базы данных. Множества для удаления дубликатов.</i>	2
	<b>*ПЗ №10.</b> Работа с текстовыми файлами. <i>Чтение и запись JSON-файла (json.load, json.dump). Парсинг структуры, преобразование в списки и словари.</i>	2
	<b>*ПЗ №11.</b> Работа с каталогами и файлами (обход, фильтрация)	2
	<b>*ПЗ №12.</b> <i>Слияние словарей.</i>	2
	<b>*ПЗ №13.</b> <i>Преобразование списка словарей в JSON.</i>	2
<b>Раздел 2. Технологии программирования</b>		<b>44</b>

Тема 2.1 Модульное программирование	<b>Содержание учебного материала</b>		<b>20</b>	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	11	<b>Подпрограммы и функции</b> Понятие подпрограммы. Процедуры и функции. Локальные и глобальные переменные. Модификаторы доступа. Передача параметров: по значению, по ссылке, по имени. <i>Функции (def) и процедуры (функции без return). Локальные и глобальные переменные (global, nonlocal). Области видимости. Передача параметров: по значению (неизменяемые типы) и по ссылке (изменяемые типы). Функции как объекты первого класса (передача функции в функцию).</i>	2	
	12	<b>Рекурсия</b> Прямая и косвенная рекурсия. Базовый случай и шаг рекурсии. Примеры: факториал, числа Фибоначчи, обход дерева. Преимущества и недостатки. <i>Рекурсивный обход каталогов (os.walk).</i>	2	
	13	<b>Модульное программирование</b> Разделение кода на модули. Интерфейс модуля. Скрытие реализации. Создание и подключение модулей. Преимущества модульного подхода. <i>Использование модуля name == «main» для тестирования.</i>	2	
	14	<b>Введение в ООП: классы и объекты</b> Базовые понятия: объект, свойства, методы, класс, интерфейс. Создание класса. Конструкторы и деструкторы. <i>Конструктор init. Атрибуты экземпляра. Методы (self).</i>	2	
	15	<b>Принципы ООП: инкапсуляция, наследование, полиморфизм</b> Инкапсуляция: сокрытие данных, геттеры/сеттеры. Наследование: база и производный класс. Полиморфизм: перегрузка методов, виртуальные функции. <i>Базовый и производный класс (super()). Полиморфизм: переопределение методов, утиная типизация (duck typing).</i>	2	
	16	<b>Системы контроля версий: Git</b> Назначение Git. Основные команды: <i>init, add, commit, push, pull, clone.</i> Понятие ветки ( <i>branch</i> ). Слияние ( <i>merge</i> ). Разрешение конфликтов.	2	
	17	<b>Отладка и профилирование</b> Инструменты отладки в Python: <i>встроенный отладчик pdb (breakpoint(), step, next, continue).</i> Логирование ( <i>модуль logging: уровни DEBUG, INFO, WARNING, ERROR, CRITICAL</i> ).	2	
	<b>в том числе, практических занятий</b>		<b>6</b>	
	*ПЗ №14. Использование подпрограмм (функции без параметров и с параметрами). <i>Создание вспомогательных функций для расчётов. Вызов функции из основной программы.</i>		2	
*ПЗ №15. Использование подпрограмм (возврат значений, переменные). <i>Функции с возвратом значений. Передача изменяемых и неизменяемых типов.</i>		2		
*ПЗ №16. Работа с рекурсивными функциями. <i>Вычисление факториала. Рекурсивный поиск вложенных элементов в списке.</i>		2		
<b>Итого за семестр</b>			<b>66</b>	
<b>СЕМЕСТР 4</b>				
Тема 2.1	<b>Содержание учебного материала</b>		<b>24</b>	ПК.2.1, ПК.2.2, ПК.2.3,
	18	<b>Функциональное программирование</b>	2	

Модульное программирование		<i>Функции map, filter, lambda-выражения. Списковые включения (list comprehensions) как альтернатива циклам. Примеры обработки коллекций.</i>		ОК.01, ОК.02, ОК.04, ОК.09
	19	<b>Исключения и управление ошибками</b> <i>Иерархия исключений. Создание собственных исключений (raise). Конструкция try-except-else-finally. Перехват нескольких типов. Использование with для ресурсов (контекстные менеджеры).</i>	2	
	<b>в том числе, практических занятий</b>		<b>20</b>	
		<b>*ПЗ №17.</b> Создание и подключение модулей. <i>Разделение кода на ES6-модули</i>	2	
		<b>*ПЗ №18.</b> Работа с классами. Создание конструкторов	2	
		<b>*ПЗ №19.</b> Применение свойств (геттеры/сеттеры). <i>Валидация данных через сеттеры</i>	2	
		<b>*ПЗ №20.</b> Работа с наследованием классов	2	
		<b>*ПЗ №21.</b> Работа с переопределением методов (полиморфизм). <i>Переопределение методов в дочерних классах. Единый интерфейс для разных типов объектов.</i>	2	
		<b>*ПЗ №22.</b> Работа с Git: <i>инициализация репозитория, коммиты, ветки. Создание и переключение веток (git branch, git checkout).</i>	2	
		<b>*ПЗ №23.</b> Разрешение конфликтов при слиянии в Git. <i>Искусственное создание конфликта и его разрешение (merge, conflict markers).</i>	2	
		<b>*ПЗ №24.</b> Выполнение отладки с pdb. <i>Установка точек останова. Пошаговое выполнение. Просмотр переменных. Логирование с модулем logging.</i>	2	
		<b>*ПЗ №25.</b> Функции map, filter и lambda-выражения. <i>Обработка списков без явных циклов. Списковые включения.</i>	2	
	<b>*ПЗ №26.</b> Обработка исключений. <i>Перехват и обработка ошибок ввода/вывода, деления на ноль, преобразования типов. Создание собственного исключения.</i>	2		
<b>Раздел 3. Разработка приложений</b>			<b>36</b>	
Тема 3.1 Этапы разработки приложений	<b>Содержание учебного материала</b>		<b>36</b>	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	20	<b>Визуально-событийное программирование</b> <i>Разработка приложений с графическим интерфейсом. Событийная модель. Компоненты интерфейса. Проектирование форм. Основы создания консольных приложений. Архитектура простого приложения: ввод → обработка → вывод. Организация меню. Многофайловая структура учебного проекта. Модуль main.py, модули с функциями и классами.</i>	2	
	21	<b>Жизненный цикл разработки приложения</b> <i>Этапы: анализ требований, проектирование, реализация, тестирование, отладка, оптимизация, сопровождение. Документирование. Жизненный цикл в контексте учебной ИС.</i>	2	
	22	<b>Проектирование объектно-ориентированного приложения</b> <i>Диаграммы классов. Диаграммы последовательностей. Выделение сущностей. Проектирование интерфейсов взаимодействия модулей. Проектирование классов «Пользователь», «Задача», «Проект».</i>	2	
	23	<b>Тестирование и отладка приложений</b>	2	

	Виды тестирования: модульное, интеграционное, системное. Тест-кейсы. Отладка. Оптимизация программы по скорости и памяти. <i>Встроенный unittest в Python. Простейшие assert. Отладка с pdb и логирование. Оптимизация: выбор структур данных (list vs dict vs set).</i>		
24	<b>Введение в сетевые взаимодействия</b> <i>Понятие API, клиент-сервер. Форматы передачи данных (JSON). Чтение данных из удалённого JSON-файла (через urllib или requests — обзорно без углубления). Обработка ответа.</i>	2	
25	<b>Безопасность и валидация.</b> <i>Валидация ввода пользователя: проверка типов, диапазонов, форматов методами строк. Основы безопасной работы с файлами. Понятие SQL-инъекции.</i>	2	
26	<b>Интеграция модулей и компонентов</b> <i>Связь между модулями через импорты. Использование функций и классов из разных файлов. Пример: модуль data.py (работа с JSON), модуль logic.py (обработка), модуль ui.py (консольный ввод/вывод).</i>	2	
27	<b>Итоговый проект: структура и этапы.</b> <i>Выбор предметной области (список контактов, задача, складской учёт). Распределение задач при разработке (индивидуально). Требования к коду, документации, тестам.</i>	2	
	<b>в том числе, практических занятий</b>	<b>20</b>	
	*ПЗ №27. Создание проекта с использованием компонентов для работы с текстом. <i>Создание формы ввода и отображение текста на веб-странице</i>	2	
	*ПЗ №28. Создание проекта с использованием кнопочных компонентов	2	
	*ПЗ №29. Создание проекта с использованием переключателей (radio, checkbox)	2	
	*ПЗ №30. Создание проекта с использованием компонентов для отображения таблиц. <i>Вывод списка записей в виде таблицы (print с выравниванием).</i>	2	
	*ПЗ №31. Создание проекта с использованием компонентов для отображения дат и времени	2	
	*ПЗ №32. Разработка интерфейса приложения. <i>Разделение на модули: main.py, models.py, storage.py, utils.py. Реализация импортов.</i>	2	
	*ПЗ №33. Разработка интерфейса приложения (привязка событий и логики)	2	
	*ПЗ №34. Тестирование приложения (написание и выполнение тест-кейсов). <i>Написание тестов для функций (валидация, сортировка, поиск). Запуск тестов. Анализ покрытия.</i>	2	
	*ПЗ №35. Создание клиента для REST API (CRUD операции)	2	
	*ПЗ №36. Создание проекта с интеграцией всех изученных тем	2	
Консультации к экзамену	<b>Содержание</b>	<b>12</b>	
	<b>1. Алгоритмизация и базовые конструкции</b> Свойства алгоритмов, способы записи, оценка сложности (O-нотация). Визуализация алгоритмов с помощью блок-схем и псевдокода. Реализация линейных, разветвляющихся и циклических алгоритмов на языке программирования. Типичные ошибки при работе с циклами и условиями (бесконечные циклы, неучтённые ветви switch). <i>Оценка сложности алгоритма обработки списка или словаря..</i>	2	ПК.2.1, ПК.2.2, ПК.2.3, ОК.01, ОК.02, ОК.04, ОК.09
	<b>2. Основы структурного программирования.</b> Одномерные и двумерные массивы: обход, поиск, вставка, удаление, сортировка. Строки: основные операции (конкатенация, поиск подстроки, замена, разбиение). Коллекции: List, Dictionary (Map), Set. Работа с текстовыми и бинарными файлами: чтение, запись,	2	

	обработка ошибок. Словари ( <i>dict</i> ): ключи и значения, итерация по парам, методы <i>get</i> , <i>keys</i> , <i>values</i> , <i>items</i> . Множества ( <i>set</i> ): уникальные элементы, операции объединения, пересечения, разности. Выбор подходящей коллекции в зависимости от задачи.		
	<b>3. Модульное программирование, рекурсия, отладка и Git</b> Подпрограммы (функции/процедуры): параметры, возвращаемые значения, область видимости. Рекурсия: базовый случай, шаг рекурсии, стек вызовов. Примеры (факториал, Фибоначчи, обход дерева). Модульное программирование: разделение кода, экспорт/импорт модулей. Отладка: точки останова, пошаговое выполнение, логирование. Git: инициализация, коммиты, ветки, слияние, разрешение конфликтов. Преобразование между JSON и структурами Python ( <i>list</i> , <i>dict</i> ). Обработка исключений: <i>try-except-else-finally</i> , перехват конкретных типов ( <i>FileNotFoundException</i> , <i>ValueError</i> , <i>KeyError</i> ). Создание собственных исключений ( <i>raise</i> ).	2	
	<b>4. Функции, рекурсия, модули, отладка</b> Классы и объекты: поля, методы, конструкторы. Инкапсуляция: модификаторы доступа, геттеры/сеттеры. Наследование: базовый и производный класс, переопределение методов. Полиморфизм: перегрузка методов, виртуальные/абстрактные методы. Git: инициализация ( <i>init</i> ), коммиты ( <i>add</i> , <i>commit</i> ), ветки ( <i>branch</i> , <i>checkout</i> ), слияние ( <i>merge</i> ), разрешение конфликтов.	2	
	<b>5. Основные принципы объектно-ориентированного программирования</b> Основы HTTP: методы (GET, POST, PUT, DELETE), коды ответа, заголовки. Форматы данных: JSON (парсинг, генерация), сравнение с XML. REST API: ресурсы, CRUD, версионирование, пагинация. Событийно-ориентированное программирование: регистрация обработчиков, всплытие событий. Полиморфизм: переопределение методов, утиная типизация ( <i>duck typing</i> ) как альтернатива классическому полиморфизму. Примеры проектирования классов для информационной системы.	2	
	<b>6. Проектирование, тестирование, безопасность и интеграция приложений</b> Этапы разработки приложений: анализ, проектирование, реализация, тестирование, оптимизация. Диаграммы классов и последовательностей (UML) для визуализации модулей. Тестирование: модульное ( <i>unit</i> ), интеграционное, системное. Написание тест-кейсов. Безопасность: XSS, CSRF, SQL-инъекции, валидация данных, JWT. Масштабируемость и производительность: кэширование, многопоточность, Web Workers. Интеграция модулей: API-шлюзы, очереди, форматы обмена.	2	
<b>Экзамен</b>		<b>6</b>	
<b>Итого за семестр</b>		<b>78</b>	
<b>ВСЕГО:</b>		<b>144</b>	

### 3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 3.1 Для реализации программы учебной дисциплины предусмотрены следующие специальные помещения:

Лаборатория «Алгоритмизации и программирования»

##### оснащена оборудованием:

- рабочее место педагога (преподавательский стол (1 шт.), стул (1 шт.));
- рабочие места обучающихся (парты ученические (13 шт.), стулья ученические (25 шт.));
- доска учебная (меловая трех-секционная) (1 шт.);
- книжный шкаф – 1 шт.;
- учебно-методическая литература по дисциплине;
- комплект учебно-наглядных пособий;

##### техническими средствами обучения:

- персональный компьютер, подключение к сети Интернет с модулем контентной фильтрации Traffic Inspector, NetPolice и YandexDNS, возможность трансляции на экран аудио и видео информации (1 шт.)
- программное обеспечение на ПК – Microsoft Windows 10 или аналог, Microsoft Office (Word, Excel, PowerPoint) или аналог, 7Zip, 24PDF, Яндекс Браузер (1 шт.)
- программное обеспечение Android Studio, Visual Studio, Visual Studio Code или аналоги
- монитор (1 шт.)
- клавиатура (1 шт.)
- мышь (1 шт.)
- телевизор (1 шт.)
- кабель для подключения HDMI (1 шт.)

#### 3.2 Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд института имеет печатные и электронные образовательные и информационные ресурсы, в том числе рекомендованные ФУМО, для использования в образовательном процессе. Список дополнен новыми изданиями.

##### 3.2.1 Основные источники

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. 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»

4. 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».

5. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Профессиональное образование). — ISBN 978-5-534-20429-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563861>

6. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 196 с. — (Профессиональное образование). — ISBN 978-5-534-18760-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584552>

7. Якимов, С. П. Алгоритмизация и программирование : учебник для среднего профессионального образования / С. П. Якимов. — Москва : Издательство Юрайт, 2026. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-19661-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589849>

### **3.2.2 Дополнительные источники**

8. Веб-разработка : учебник для среднего профессионального образования / под общей редакцией О. В. Ратановой, Н. А. Ребус, А. Ю. Анисимова. — Москва : Издательство Юрайт, 2026. — 217 с. — (Профессиональное образование). — ISBN 978-5-534-21227-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590627>

9. 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>

10. 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>

11. 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 PROФобразование : [сайт]. — URL: <https://profspo.ru/books/122426>

12. 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>

13. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебник для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2026. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587050>

14. Полуэктова, Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-18644-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588469>

15. Федоров, Д. Ю. Программирование на Python : учебник для среднего профессионального образования / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 166 с. — (Профессиональное образование). — ISBN 978-5-534-22180-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/600873>

16. Черпаков, И. В. Алгоритмизация и программирование в Python : учебник для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2026. — 159 с. — (Профессиональное образование). — ISBN 978-5-534-21911-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582413>

### **3.2.3. Электронные источники:**

17. Сайт по программированию <https://metanit.com>

## 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Методы оценки
<p>Уметь:</p> <ul style="list-style-type: none"> <li>– распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части;</li> <li>– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации;</li> <li>– взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности;</li> <li>– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы;</li> <li>– разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий;</li> <li>– анализировать требования и определять функциональность модуля;</li> <li>– создавать интерфейсы для взаимодействия с другими модулями и системами;</li> <li>– обеспечивать безопасность, производительность и масштабируемость при разработке модулей;</li> <li>– оптимизировать проектируемые модули для повышения их эффективности и качества;</li> <li>– определять структуру и интерфейсы модулей;</li> <li>– анализировать требования к модулю и определять его функциональность;</li> <li>– создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля;</li> </ul>	<p><b>Оценка «5»:</b>  умет глубоко анализировать задачу, выделять скрытые составляющие и неявные требования, строить причинно-следственные связи. Способен самостоятельно определять задачи поиска информации, критически оценивать источники, адаптировать стратегию под контекст. Взаимодействует продуктивно, управляет коммуникацией, разрешает разногласия. Понимает детали, нюансы и подтекст профессиональных текстов и высказываний. <i>Разрабатывает консольные приложения на Python с многоуровневым меню, подсказками и валидацией ввода. Использует все типы коллекций (list, dict, set, tuple) и строковые методы, выбирая оптимальную структуру данных. Применяет все управляющие конструкции (if-elif-else, for, while, break, continue), создаёт функции с параметрами и возвращаемыми значениями, корректно использует рекурсию (факториал, Фибоначчи, обход вложенных структур). Выполняет чтение и запись текстовых файлов и JSON с обработкой ошибок и использованием контекстного менеджера with. Создает классы с конструктором, инкапсуляцией (property), наследованием (super) и полиморфизмом. Применяет продвинутые команды Git (rebase, cherry-pick, stash), разрешает сложные конфликты. Отлаживает код с помощью pdb (условные точки останова) и многоуровневого логирования (logging). Обработывает исключения с конкретными типами (ValueError, TypeError, FileNotFoundError), создаёт собственные исключения. Уверенно использует lambda, map, filter и списковые включения для сложных сценариев. Разрабатывает законченное консольное приложение с тестами (unittest), документацией и хранением данных в JSON.</i></p> <p><b>Оценка «4»:</b>  Студент умеет структурировать задачу, выделять ключевые и второстепенные части, видеть простые взаимосвязи. Способен разбивать поиск на этапы, обосновывать выбор источников, использовать разные типы источников. Взаимодействует активно, аргументирует позицию, учитывает мнение других. Понимает ключевую информацию из профессиональных текстов, может пересказать основное содержание.</p>	<p><b>Текущий контроль:</b>  Оценка результатов выполнения практической работы  Экспертное наблюдение за ходом выполнения практической работы (деятельностью студента)  Оценка выполнения практического задания (работы)  <b>Промежуточная аттестация:</b> экзамен</p>

<ul style="list-style-type: none"> <li>– проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами;</li> <li>– учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля;</li> <li>– интегрировать модули и компоненты, обеспечивая их взаимодействие;</li> <li>– работать с различными форматами данных и протоколами передачи данных;</li> <li>– <i>разрабатывать консольные приложения на Python с меню и обработкой ввода;</i></li> <li>– <i>использовать списки, словари, множества и строки для обработки данных;</i></li> <li>– <i>применять ветвления, циклы, функции и рекурсию при решении алгоритмических задач;</i></li> <li>– <i>выполнять чтение и запись текстовых файлов и JSON;</i></li> <li>– <i>создавать классы и объекты с инкапсуляцией, наследованием и полиморфизмом;</i></li> <li>– <i>использовать Git (коммиты, ветки, слияние) для управления проектами;</i></li> <li>– <i>отлаживать код с помощью pdb и логирования;</i></li> <li>– <i>обрабатывать исключения (try-except);</i></li> <li>– <i>применять lambda, map, filter и списковые включения; разрабатывать законченное консольное приложение с тестами (unittest) и хранением данных в JSON.</i></li> </ul>	<p><i>Разрабатывает консольные приложения на Python с меню из 3–5 пунктов. Использует 2–3 типа коллекций (list, dict, set) и базовые строковые методы. Применяет базовые ветвления и циклы, создаёт функции с параметрами, использует рекурсию для типовых задач (факториал, Фибоначчи). Выполняет чтение и запись текстовых файлов и JSON, обрабатывает основные ошибки (файл не найден, некорректный JSON). Создает классы с конструктором и методами, применяет базовое наследование. Использует базовые команды Git (init, add, commit, branch, checkout, merge), разрешает простые конфликты. Отлаживает код с помощью пошагового выполнения pdb и просмотра переменных. Обрабатывает несколько типов исключений (ValueError, FileNotFoundError). Применяет lambda и map/filter для простых случаев, использует базовые списковые включения. Разрабатывает законченное консольное приложение с базовым меню, простыми тестами (2–3 assert) и сохранением в JSON.</i></p> <p><b>Оценка «3»:</b>  Студент умеет выделить наличие задачи, назвать 1–2 очевидных составляющих. Способен сформулировать простой запрос, назвать 1–2 источника информации. Взаимодействует на базовом уровне, отвечает на вопросы, следует заданной роли. Понимает основную тему профессионального текста, отдельные знакомые термины. Разрабатывает простейшие консольные приложения на Python (10–20 строк) без обработки ошибок. Использует один структурированный тип (например, список) и 1–2 строковых метода. Применяет простые ветвления (if-else) и циклы (for или while), может написать простую функцию без параметров. Выполняет чтение или запись одного формата (текстовый файл или JSON) без обработки ошибок. Создает простой класс с конструктором и 1–2 методами, без наследования. Использует базовый цикл Git (init, add, commit). Отлаживает код с помощью точек останова в IDE или print(). Использует try-except без уточнения типов исключений. Применяет простую lambda-функцию или базовое списковое включение. Разрабатывает простейшее консольное приложение без тестов, с одноразовым сохранением в JSON.</p>	
---	---	--

<p>Знать:</p> <ul style="list-style-type: none"> <li>– актуальный профессиональный и социальный контекст, в котором приходится работать и жить;</li> <li>– номенклатура информационных источников, применяемых в профессиональной деятельности;</li> <li>– язык программирования, основные конструкции, синтаксис;</li> <li>– структуры данных;</li> <li>– принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP;</li> <li>– работа с инструментальным программным обеспечением;</li> <li>– многопоточность в программных модулях;</li> <li>– основные принципы проектирования модулей программного обеспечения;</li> <li>– языки программирования и технологии для реализации модулей;</li> <li>– принципы создания интерфейсов для взаимодействия с другими модулями и системами;</li> <li>– принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей;</li> <li>– методы и подходы к интеграции модулей и компонентов;</li> <li>– принципы безопасности при интеграции модулей и компонентов.</li> <li>– свойства алгоритмов, способы записи, основы О-нотации;</li> <li>– особенности Python: динамическая типизация, изменяемые/неизменяемые типы;</li> <li>– синтаксис Python: переменные, операторы, ветвления, циклы, ввод/вывод;</li> </ul>	<p><b>Оценка «5»:</b></p> <p>ориентируется в текущей ситуации отрасли (востребованные технологии, зарплаты, карьерные траектории, этические и правовые аспекты). системно классифицирует источники, знает авторитетные ресурсы для разных задач (MDN, GitHub, Habr, Tproger, курсы), умеет оценивать достоверность. глубокое знание синтаксиса и семантики, включая современные возможности (стрелочные функции, деструктуризация, spread/rest, шаблонные строки и др.). знает очередь, стек, связные списки, деревья, хеш-таблицы; понимает их сложность и выбор под задачу. сравнивает REST и SOAP, знает альтернативы (GraphQL, gRPC), понимает версионирование, документацию (Swagger/OpenAPI). ориентируется в спектре инструментов (сборщики, линтеры, тестовые фреймворки, профилировщики, CI/CD). знает модели многопоточности в разных языках, понимает асинхронность как альтернативу, знает Web Workers. объясняет каждый принцип SOLID с примерами, знает принципы связности и сцепления, паттерны проектирования. ориентируется в экосистеме языков, знает сильные и слабые стороны каждого для разных задач, следит за новыми технологиями. объясняет OWASP Top 10, профилирование производительности, паттерны масштабирования (репликация, шардирование, микросервисы). знает подходы: точка-точка, шина данных, сервисная сетка (service mesh), очереди сообщений (RabbitMQ, Kafka). объясняет безопасность межмодульного взаимодействия.</p> <p><i>особенности Python: динамическую типизацию, изменяемые/неизменяемые типы;</i></p> <p><i>весь синтаксис Python (переменные, операторы, ветвления, циклы, ввод/вывод); все структурированные типы: list, dict, set, tuple;</i></p> <p><i>работу с файлами (open, with) и JSON (load, dump, loads, dumps);</i></p> <p><i>модульное программирование (import, from...import, if name == «main»);</i></p> <p><i>рекурсию (базовый случай, шаг, стек вызовов);</i></p> <p><i>основы ООП: классы, объекты, init, наследование, полиморфизм, инкапсуляцию (property);</i></p>	<p><b>Текущий контроль:</b> Компьютерное тестирование по основным разделам программы. Решение задач по основным разделам программы</p> <p><b>Промежуточная аттестация:</b> экзамен</p>
--	--	--

<ul style="list-style-type: none"> <li>– структурированные типы: <i>list, dict, set, tuple</i>;</li> <li>– работа с файлами (<i>open, with</i>) и JSON (<i>json.load, json.dump</i>);</li> <li>– модульное программирование (<i>import, if name == «main»</i>);</li> <li>– рекурсия (базовый случай, шаг рекурсии);</li> <li>– основы ООП: классы, объекты, <i>init</i>, наследование, полиморфизм, инкапсуляция (<i>property</i>);</li> <li>– основы Git (<i>init, add, commit, branch, merge</i>);</li> <li>– отладка (<i>pdb</i>) и логирование (<i>logging</i>);</li> <li>– функциональные возможности: <i>lambda, map, filter, list comprehensions</i>;</li> <li>– исключения (<i>try-except-else-finally</i>) и контекстные менеджеры (<i>with</i>);</li> <li>жизненный цикл приложения, основы тестирования (<i>unittest</i>), валидация ввода.</li> </ul>	<p><i>основы Git (init, add, commit, branch, merge, rebase, stash); отладку (pdb) и логирование (logging с уровнями); lambda, map, filter, list comprehensions; исключения (try-except-else-finally) и контекстные менеджеры (with); жизненный цикл приложения, основы unittest, валидацию ввода.</i></p> <p><b>Оценка «4»:</b></p> <p>понимает современные тренды разработки, требования рынка, особенности удалённой/офисной работы. называет основные типы источников (официальная документация, профильные форумы, блоги, видеоуроки, книги). уверенно называет все ключевые конструкции (функции, массивы, объекты, обработку ошибок, модули). знает список, словарь, множество, понимает различия и области применения. называет основные инструменты (редактор кода, отладчик, система контроля версий, терминал). объясняет проблемы многопоточности (гонки данных, дедлоки), базовые механизмы синхронизации. называет SOLID, DRY, KISS. называет 2–3 языка и их типичные применения (бэкенд, фронтенд, скрипты). называет конкретные меры: валидация, кэширование, горизонтальное масштабирование. называет событийно-ориентированную интеграцию, API-шлюзы. называет аутентификацию, авторизацию, валидацию входящих данных, защиту от подделки запросов. динамическую типизацию Python; основной синтаксис Python (переменные, операторы, ветвления, циклы, ввод/вывод); list и dict (set/tuple — обзорно); работу с файлами (open, with) и JSON (load, dump); основы модульного программирования (import, if name == «main»); понятие рекурсии (базовый случай и шаг) основы ООП (классы, объекты, init, наследование); основы Git (init, add, commit, branch, merge); отладку (pdb) и логирование (logging); try-except и обзорно with; основы жизненного цикла, unittest, валидации; общие принципы создания интерфейсов, работы с инструментальным ПО;</p>	
---	--	--

	<p><i>общие принципы проектирования, безопасности, производительности, интеграции.</i></p> <p><b>Оценка «3»:</b></p> <p>знает общие черты профессии (разработка ПО, работа в команде).  знает 1–2 источника (например, Stack Overflow, документацию).  знает основные операторы, типы данных, базовый синтаксис (условия, циклы).  знает массивы и строки.  знает, что такое API, называет REST.  знает среду разработки (IDE) и базовые возможности.  знает понятие потока/процесса.  знает принцип разделения на модули.  знает один язык (например, Python).  знает термины «безопасность», «производительность», «масштабируемость».  знает прямой вызов одного модуля другим;  знает про аутентификацию;  <i>отдельные элементы профессионального контекста, называет 1–2 источника; отдельные свойства алгоритмов (конечность, детерминированность); знает отдельные конструкции Python (переменные, if, for/while, input/print); понятие списка (list) и словаря (dict); знает о существовании файлов и JSON; знает, что код можно разделять на модули; знает, что функция может вызывать саму себя (рекурсия); понятие класса и объекта; базовые команды Git (init, add, commit); о существовании отладки и логирования; понятие lambda (короткая функция); о существовании try-except; понятие жизненного цикла, тестирования, валидации; отдельные термины по интерфейсам, инструментальному ПО, проектированию, безопасности, интеграции</i></p>	
--	---	--