ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ИННОПРОГ»

Проект

УТВЕРЖДАЮ

Генеральный директор ООО «ИННОПРОГ»



_____ / Венедиктов Р.В.

подпись инициалы, фамилия

4 сентября 2026 г.

Приказ об утверждении программы

от 4 сентября 2026 г. № ОБР-17

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Python-разработчик для детей»

Направленность:	техническая
Уровень освоения:	базовый
Возраст обучающихся:	9–17 лет
Трудоёмкость:	160 академических часов
Форма обучения:	очная, с применением электронного обучения и дистанционных образовательных технологий
Режим реализации:	80 учебных недель
Язык обучения:	русский

Иннополис, 2026

Содержание

1 Пояснительная записка . . . 3
1.1 Наименование, вид и направленность программы . . . 3
1.2 Актуальность и педагогическая целесообразность . . . 3
1.3 Адресат, входные требования и диагностика . . . 3
1.4 Порядок приёма, сопровождения и прекращения обучения . . . 4
1.5 Форма, режим и срок освоения . . . 4
1.6 Цель, задачи и планируемые результаты . . . 5
1.7 Нормативные основания . . . 6
1.8 Структура программы и документ об обучении . . . 6

2 Содержание программы . . . 7
2.1 Учебный план . . . 7
2.2 Календарный учебный график . . . 8
2.3 Рабочие программы модулей . . . 9
2.4 Проектная практика и консультации . . . 44

3 Организационно-педагогические условия . . . 44
3.1 Общие условия реализации . . . 44
3.2 Кадровое и методическое обеспечение . . . 45
3.3 Оборудование и учебная среда . . . 45
3.4 Электронная информационно-образовательная среда . . . 45
3.5 Здоровьесбережение, доступность и цифровая безопасность . . . 46

4 Оценка качества освоения программы . . . 46
4.1 Входная диагностика и текущий контроль . . . 46
4.2 Практикумы и обратная связь . . . 47
4.3 Внутренняя оценка качества реализации . . . 47

5 Методические материалы . . . 47
5.1 Указания для обучающихся . . . 47
5.2 Указания для преподавателя . . . 47
5.3 Литература и электронные ресурсы . . . 48

Приложение 1. Матрица формирования результатов . . . 49

Приложение 2. Практикумы и обратная связь . . . 50

Приложение 3. Учебные проекты . . . 51

Приложение 4. Памятка по подготовке работы . . . 53

Приложение 5. Лист актуализации программы . . . 54

1. Пояснительная записка

1.1. Наименование, вид и направленность программы

Дополнительная общеобразовательная общеразвивающая программа «Python-разработчик для детей».

Дополнительная общеобразовательная общеразвивающая программа технической направленности. Подвид образования — дополнительное образование детей и взрослых. Адресат данной программы — несовершеннолетние в указанном возрастном диапазоне. Уровень освоения — базовый. Язык образования — русский. Предыдущая редакция утверждена приказом ООО «ИННОПРОГ» от 05.08.2026 № ОБР-13. Возрастной диапазон расширен с 9–14 до 9–17 лет. Форма обучения — очная. Занятия и контрольные мероприятия реализуются дистанционно, посредством электронного обучения и дистанционных образовательных технологий. Основная форма взаимодействия — индивидуальные синхронные занятия с преподавателем.

1.2. Актуальность и педагогическая целесообразность

Программа формирует основы программирования через короткие задания с наблюдаемым результатом и постепенно переводит обучающегося к самостоятельной разработке. Python используется для создания небольших программ, которые принимают ввод, обрабатывают данные и сохраняют результат. Проектная практика помогает связать написание кода с проверкой и объяснением его работы.

Практическое время нужно для перехода от небольших выражений к работе с файлами, обработке ошибочного ввода и проекту, который можно запустить повторно. Базовые темы дают опору младшим, а более подготовленные учащиеся используют то же время для расширенных сценариев, тестирования и доработки.

1.3. Адресат, входные требования и диагностика

Возраст обучающихся — 9–17 лет. Предварительные знания Python не требуются. На вводной диагностике обучающийся открывает файл, набирает

короткий текст и по образцу запускает программу. При необходимости преподаватель показывает эти действия и предоставляет подготовленную среду.

Для обучающихся 9–11 лет преподаватель использует короткие инструкции, поэтапные примеры и заготовки с видимым результатом. В 12–14 лет обучающийся чаще самостоятельно выбирает структуры данных и добавляет проверки. В 15–17 лет увеличиваются самостоятельность декомпозиции и число граничных сценариев. Возраст служит ориентиром, конкретную опору определяют по диагностике. При отсутствии графической среды готовую учебную среду предоставляет преподаватель. Публикация проекта и подключение мессенджера не требуются для освоения программы.

Перед началом занятий преподаватель предлагает короткое задание, выясняет интересы обучающегося и фиксирует исходный уровень. С учётом результатов подбираются объём опор, сложность и примеры. Индивидуальное расписание определяет даты занятий, сохраняя трудоёмкость, обязательные результаты и предусмотренные программой формы контроля. Зачисление оформляется в порядке, установленном локальными актами организации.

1.4. Порядок приёма, сопровождения и прекращения обучения

Приём, зачисление и прекращение обучения оформляются по локальным актам организации. При работе с несовершеннолетними применяются предусмотренные организацией процедуры взаимодействия с законным представителем. Преподаватель фиксирует учебную цель, согласованный маршрут и результаты диагностики в принятой форме учёта. При пропуске занятия согласуются порядок восполнения и форма проверки результата.

1.5. Форма, режим и срок освоения

Программа рассчитана на 160 академических часов за 80 учебных недель. Занятия планируются раздельными встречами с учётом возраста, продолжительности экранной работы и перерывов. Индивидуальный учебный план и календарный график оформляются по локальному порядку. При индивидуальном графике с семьёй может быть согласован более продолжительный календарный срок с сохранением 160 академических часов, обязательных результатов и аттестации. До зачисления фиксируются программа, плановый срок, режим

и стоимость по применимым договорным документам. Месячный абонемент покрывает согласованное количество занятий за расчётный период.

1.6. Цель, задачи и планируемые результаты

Научить проектировать, реализовывать, проверять и представлять небольшие программы на Python, объясняя связь между исходными данными, алгоритмом и результатом.

Учебные задачи — последовательно сформировать перечисленные умения и отработать их в практических заданиях. Развивающие задачи — научить планировать решение, проверять гипотезу на примерах и исправлять ошибки. Воспитательные задачи — формировать самостоятельность, ответственное обращение с цифровыми материалами и уважение к авторству.

По завершении программы обучающийся выполняет следующие действия.

– Р1. Создаёт и запускает файл Python, читает простой код и объясняет порядок выполнения команд.
– Р2. Выбирает тип данных, формулирует условия, применяет циклы и отслеживает изменение значений на контрольных примерах.
– Р3. Организует данные в списках, словарях и множествах, выполняет отбор, поиск, подсчёт и сортировку.
– Р4. Разбивает решение на функции, применяет стандартные модули и различает локальные переменные, параметры и возвращаемые значения.
– Р5. Читает и сохраняет учебные данные в текстовом формате, CSV и JSON, проверяет структуру входных данных.
– Р6. Различает синтаксическую ошибку, исключение и логический дефект, воспроизводит сбой и проверяет исправление.
– Р7. Моделирует состояние объекта или диалога и обновляет его по действиям пользователя.
– Р8. Представляет завершённый мини-проект, инструкцию запуска и протокол проверок.

1.7. Нормативные основания

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», в том числе статьи 2, 12, 16, 28, 29, 54, 58, 60 и 75.
- Приказ Минпросвещения России от 27.07.2022 № 629, утвердивший Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам.
- Постановление Правительства Российской Федерации от 11.10.2023 № 1678 о применении электронного обучения и дистанционных образовательных технологий в редакции постановления от 06.05.2026 № 517.
- СП 2.4.2.4283-26, утверждённые постановлением Главного государственного санитарного врача Российской Федерации от 02.06.2026 № 19, и СанПиН 1.2.3685-21 в применимой редакции, включая таблицу 6.8 о продолжительности использования электронных средств обучения.
- Устав, правила приёма и положения ООО «ИННОПРОГ» об образовательном процессе, режиме занятий, электронном обучении и дистанционных технологиях, текущем контроле, промежуточной и итоговой аттестации, порядке выдачи документов об обучении. Приказ ООО «ИННОПРОГ» от 02.08.2026 № ОБР-12.

1.8. Структура программы и документ об обучении

В комплект входят учебный план, календарный учебный график, рабочие программы модулей, организационно-педагогические условия, оценочные и методические материалы. Освоение программы фиксируется по результатам текущего контроля. При освоении программы и выполнении предусмотренных учебным планом работ ООО «ИННОПРОГ» оформляет сертификат как документ об обучении по установленному организацией образцу. Лицам, освоившим часть программы или отчисленным из организации, выдаётся справка об обучении или о периоде обучения в соответствии с частью 12 статьи 60 Федерального закона № 273-ФЗ.

2. Содержание программы

2.1. Учебный план

Академический час составляет 45 минут и служит единицей учёта трудоёмкости. На странице курса длительность индивидуального урока указана в пределах 45–60 минут, а расписание согласуется индивидуально. Плановый объём 160 академических часов и нормативный учебный график определены этой программой. Перерывы и организационные паузы в учебные часы не включаются. Тематические часы включают объяснение, демонстрацию, практические задания и текущий контроль. Отдельный раздел «Практика» и направление на внешнюю базу практики учебным планом не предусмотрены. Учебные проекты и практикумы входят в часы соответствующих модулей. Преподаватель проверяет их в рамках текущего контроля. Дополнительные упражнения на закрепление, которые педагог предлагает по желанию, в обязательную трудоёмкость не входят.

Таблица 1 – Учебный план

№	Учебный модуль	Теория	Практика	Всего
1	Последовательность команд и запуск программы	2	4	6
2	Переменные и диалог с пользователем	2	4	6
3	Числа и вычисления	2	4	6
4	Обработка строк	2	4	6
5	Условия и выбор действий	2	4	6
6	Цикл while и условие завершения	2	4	6
7	Цикл for и перебор	2	8	10
8	Списки и изменение коллекций	2	4	6
9	Поиск, подсчёт и сортировка	2	4	6
10	Словари и множества	2	4	6
11	Функции и декомпозиция	2	4	6
12	Стандартные модули и случайные события	2	4	6
13	Текстовые файлы и сохранение	2	5	7
14	Таблицы CSV и данные JSON	2	5	7

Продолжение таблицы 1

№	Учебный модуль	Теория	Практика	Всего
15	Диагностика ошибок и надёжный ввод	2	9	11
16	Объекты и состояние	2	5	7
17	Рисование и события	2	6	8
18	Диалоговый помощник	2	6	8
19	Планирование и разработка мини-проекта	3	11	14
20	Проверка, доработка и представление проекта	2	20	22
	Итого	41	119	160

2.2. Календарный учебный график

Недели отсчитываются от начала обучения. Если темы сменяются внутри недели, она указана в обеих строках. Часы в таблице относятся ко всей теме. Даты занятий и перерывы определяются индивидуальным расписанием.

Таблица 2 – Календарный учебный график

Учебный модуль	Учебные недели	Часы	Результат работы
Последовательность команд и запуск программы	1–3	6	Запущенный файл с последовательностью команд
Переменные и диалог с пользователем	4–6	6	Диалог с пользователем
Числа и вычисления	7–9	6	Расчёт с проверкой типов
Обработка строк	10–12	6	Обработка текста по правилам
Условия и выбор действий	13–15	6	Ветвление на контрольных примерах
Цикл while и условие завершения	16–18	6	Цикл с проверкой остановки
Цикл for и перебор	19–23	10	Отчёт по последовательности
Списки и изменение коллекций	24–26	6	Программа со списком
Поиск, подсчёт и сортировка	27–29	6	Поиск и сортировка с тестами
Словари и множества	30–32	6	Справочник на словарях и множествах

Продолжение таблицы 2

Учебный модуль	**Учебные недели**	**Часы**	**Результат работы**
Функции и декомпозиция	33–35	6	Решение из нескольких функций
Стандартные модули и случайные события	36–38	6	Программа со стандартным модулем
Текстовые файлы и сохранение	39–42	7	Чтение и запись текста
Таблицы CSV и данные JSON	42–45	7	Обмен данными в CSV и JSON
Диагностика ошибок и надёжный ввод	46–51	11	Исправленный сценарий ошибочного ввода
Объекты и состояние	51–54	7	Небольшая модель на классах
Рисование и события	55–58	8	Интерактивная графика
Диалоговый помощник	59–62	8	Диалоговый помощник
Планирование и разработка мини-проекта	63–69	14	Первая версия собственного проекта
Проверка, доработка и представление проекта	70–80	22	Рабочий проект и инструкция запуска
Итого	80	160	

2.3. Рабочие программы модулей

Тематические таблицы показывают последовательность учебных действий. Указанные часы темы включают разбор и выполнение упражнения, а также проверку результата. Типовые ошибки используются для обратной связи и выбора повторного задания. Контрольная работа модуля входит в практические часы.

Модуль 1. Последовательность команд и запуск программы

Предметный результат. Открывает, сохраняет и запускает файл, прогнозирует порядок вывода.

Содержание. Алгоритм описывает порядок действий исполнителя. Текст исходного файла читается человеком, а команды исполняет интерпретатор. В редакторе различаются рабочая папка, открытый файл и консоль. Функция print

выводит значения, комментарий поясняет замысел. По выводу удобно прослеживать порядок выполнения.

Практическая работа. Записать алгоритм подготовки к занятию, обнаружить зависимые действия. Создать и сохранить файл, вывести три строки, переставить две команды и сравнить результат. Исправить незакрытую кавычку, указанную интерпретатором.

Таблица 3 – Тематический план модуля 1

№	Тема	Часы	Содержание и учебное действие
1	Команда и исполнитель	1	Команда указывает исполнителю отдельное действие. В последовательной программе порядок команд определяет порядок наблюдаемых действий. Практика. Переставить два действия в цепочке «открыть, прочитать, закрыть» и отметить, какой порядок выполним.
2	Бытовой алгоритм	1	В алгоритме важно уточнять начало, конец и зависимости. Если выбрать действие раньше получения нужной информации, инструкция становится неоднозначной. Практика. Записать шесть действий для сборов к уроку и поручить товарищу или преподавателю проверить последовательность по записи.
3	Файл и редактор	1	Редактор изменяет текст исходника, который хранится в файле с расширением .py. Для исходника выбирают рабочую папку, чтобы при повторном запуске открыть нужную версию. Практика. Создать папку и файл hello.py. Закрыть редактор, снова открыть именно этот файл и показать его путь.
4	Сохранение и выполнение	1	Интерпретатор выполняет сохранённый код выбранного файла. Изменение в несохранённой вкладке и вывод предыдущего запуска удобно различать по специально добавленной строке. Практика. Добавить отличимую строку, сохранить и запустить. Сопоставить имя файла и вывод.
5	Вывод через print	1	Функция print выводит переданные значения. Несколько вызовов позволяют наблюдать выполнение команд по шагам. Практика. Вывести имя вымышленного персонажа и две его реплики отдельными строками. Объяснить порядок.

Продолжение таблицы 3

№	Тема	Часы	Содержание и учебное действие
6	Чтение кода и комментарии	1	Комментарий после знака # поясняет человеку назначение фрагмента. При чтении чужого кода полезно сначала предсказать результат, затем проверить прогноз запуском. Практика. Объяснить программу из четырёх строк, отметить строки вывода и добавить комментарий к её назначению.
	Итого	6	

Типовые ошибки и способы проверки.

– Редактор показывает новый текст, а консоль выводит прежний. Сохранить файл и сверить путь команды запуска.
– Интерпретатор указал на строку вывода. Проверить закрытие кавычек и скобок в этой и предыдущей строке.

Текущий контроль. Написать приветствие из трёх команд, добавить поясняющий комментарий и показать изменение результата после перестановки строк.

Предъявляемые материалы. Файл .py и два снимка или текстовых протокола запуска.

Ожидаемый результат. Вывод совпадает с порядком команд и не включает текст комментария.

Модуль 2. Переменные и диалог с пользователем

Предметный результат. Сохраняет ответы и составляет сообщение из введённых значений.

Содержание. Имя переменной связывается со значением. При повторном присваивании последующее чтение имени даёт новое значение. input возвращает строку, её можно объединять с текстом. F-строка позволяет вставить значение в сообщение. Имена переменных подбираются по роли данных.

Практическая работа. Сделать анкету об увлечениях с вымышленными ответами и персональным приветствием. Переписать часть сообщений через f-строки, проверить два набора ввода и вывести подпись с текущим значением переменной.

Таблица 4 – Тематический план модуля 2

№	Тема	Часы	Содержание и учебное действие
1	Связывание имени со значением	1	Оператор присваивания сначала вычисляет правую часть и связывает результат с именем слева. При печати имени получается связанное с ним значение. Практика. Записать имя персонажа в переменную и использовать его в двух репликах.
2	Имена и повторное присваивание	1	Имена отражают роль значения, например hobby обозначает увлечение. Повторное присваивание изменяет значение, которое будет получено при следующем обращении к имени. Практика. Заполнить таблицу значений до и после присваивания и подтвердить её двумя вызовами print.
3	Строки и кавычки	1	Строковый литерал ограничивается согласованными кавычками. Внутреннюю кавычку можно представить экранированием или выбрать другой тип внешних кавычек. Практика. Вывести реплику с кавычками внутри, затем исправить специально подготовленный строковый литерал.
4	Ввод с клавиатуры	1	input показывает подсказку и возвращает введённый текст как строку. Пустая отправка даёт пустую строку, что важно для будущих проверок. Практика. Запросить вымышленное имя и увлечение. Показать, что числовые символы в ответе также приходят как текст.
5	Соединение строк	1	Оператор + соединяет две строки без автоматической вставки пробела. Пробел и знаки препинания входят в формируемый текст. Практика. Собрать приветствие из трёх частей и исправить место, где слова сливаются.
6	Подстановка через f-строку	1	Выражение в фигурных скобках вычисляется в момент формирования строки. Один шаблон даёт разные реплики в зависимости от входных значений. Практика. Заменить ручное соединение f-строкой и сравнить оба варианта на двух одинаковых входах.
	Итого	6	

Типовые ошибки и способы проверки.

– На экране появилось слово name вместо ответа. Проверить, передано ли значение переменной, и есть ли префикс f перед строкой с фигурными скобками.
– Поздний ответ заменяет первый. Для независимых полей завести отдельные имена.

Текущий контроль. Сделать учебную анкету с двумя ответами и персональным сообщением.

Предъявляемые материалы. Код и вывод двух запусков с разными вымышленными данными.

Ожидаемый результат. Оба значения выведены в нужном месте, а данные одного запуска не подставляются в другой.

Модуль 3. Числа и вычисления

Предметный результат. Выбирает тип и операцию под условие, проверяет расчёт вручную.

Содержание. Целые и вещественные числа, преобразование строк в числа, приоритет операций и скобки. Операторы /, // и % решают разные задачи. Округление представления результата отличается от исходного значения. До вычислений по образцу проверяются допустимые исходные значения.

Практическая работа. Сделать калькулятор стоимости нескольких позиций и перевод продолжительности в часы и минуты. Вручную вычислить ответ для граничных значений 0, 59 и 60 минут. Сопоставить вывод для целой и дробной цены.

Таблица 5 – Тематический план модуля 3

№	Тема	Часы	Содержание и учебное действие
1	Целые и дробные числа	1	Тип int хранит целые числа, a float представляет вещественные значения с ограниченной точностью. Количество предметов удобно хранить целым, длину в метрах можно представить дробным значением. Практика. Классифицировать четыре величины и вывести типы двух присвоенных значений.

Продолжение таблицы 5

№	Тема	Часы	Содержание и учебное действие
2	Преобразование ввода	1	Числовой текст превращают в int или float до арифметической операции. Формат десятичной точки указывают в подсказке, а при неверном формате интерпретатор сообщит об ошибке. Практика. Проверить сложение двух вводов с преобразованием и без него. Отдельно показать ввод дробного числа с точкой.
3	Основные арифметические операции	1	Сложение и вычитание изменяют величину, умножение применяет одинаковый множитель, деление распределяет значение. При делении нужно исключить нулевой делитель. Практика. Рассчитать стоимость трёх одинаковых предметов и долю одного участника при равном распределении ненулевой суммы.
4	Частное и остаток	1	Для неотрицательной длительности // находит целое число единиц, а % остаток. Проверка через произведение частного на делитель и добавление остатка восстанавливает исходное значение. Практика. Преобразовать 125 минут и подтвердить равенство 125 = 2 × 60 + 5.
5	Скобки и порядок операций	1	Умножение и деление имеют приоритет перед сложением. Скобки показывают, какое подвыражение требуется вычислить раньше. Практика. Сравнить результаты 2 + 3 × 4 и (2 + 3) × 4, затем проверить соответствующие выражения Python.
6	Точность и ручная проверка	1	Округление отображаемого ответа и изменение самого значения — отдельные решения. Ручной расчёт одного простого входа помогает обнаружить ошибку в формуле. Практика. Подготовить расчёт для 3 единиц по 12.5 и отобразить результат с двумя знаками после точки.
	Итого	6	

Типовые ошибки и способы проверки.

– Сложение двух значений ввода склеило символы. Преобразовать каждое значение к числу до вычисления.

– Длительность 125 минут показана как 2 часа 2 минуты. Для минут взять остаток, а часы получить целочисленным делением.

Текущий контроль. Получить часы и минуты из неотрицательной целой длительности, затем проверить вычисления.

Предъявляемые материалы. Код и таблица для 0, 59, 60 и 125 минут.

Ожидаемый результат. Результаты 0:00, 0:59, 1:00 и 2:05, допускаются равнозначные форматы вывода.

Модуль 4. Обработка строк

Предметный результат. Применяет индексы, срезы и операции нормализации строки.

Содержание. Строка является последовательностью символов. Индекс обращается к одному элементу, срез извлекает диапазон, правая граница исключается. При поиске учитывается регистр. Методы обработки возвращают новые строки, пробелы по краям можно удалить отдельно.

Практическая работа. Очистить введённое имя, извлечь часть строки и посчитать число слов в короткой фразе. Подготовить текст с пробелами и разным регистром и сравнить поиск до и после нормализации.

Таблица 6 – Тематический план модуля 4

№	Тема	Часы	Содержание и учебное действие
1	Длина строки	1	len сообщает число элементов строки, включая пробелы. Пустая строка имеет длину ноль, а строка из пробелов становится пустой после удаления краевых пробелов. Практика. Сравнить длину обычного текста, пустого ввода и двух пробелов до и после strip.
2	Индексы и границы	1	Первый элемент имеет индекс 0. Последний неотрицательный индекс на единицу меньше длины, а отрицательные индексы отсчитываются от конца. Практика. Вывести первый и последний символ непустого текста и указать, почему индекс len недопустим.

Продолжение таблицы 6

№	Тема	Часы	Содержание и учебное действие
3	Срезы	1	Срез возвращает диапазон элементов, не включая правую границу. Он удобен для извлечения префикса или краткого фрагмента. Практика. Из строки «проект» получить «про» и «ект» разными срезами, подписав границы.
4	Поиск части строки	1	Выражение in проверяет вхождение последовательности символов. Для отдельного слова следует сначала выделить слова по выбранному правилу. Практика. Показать разницу между поиском «кот» как подстроки и как элемента списка в строке «котик и кот».
5	Регистр и пробелы	1	strip удаляет пробелы по краям, a lower приводит буквы к нижнему регистру. Эти операции возвращают новые строки, поэтому результат нужно сохранить или сразу использовать. Практика. Сопоставить « Привет » и «привет» после одинаковой нормализации.
6	Разбиение и сборка	1	split выделяет части текста по разделителю или пробельным символам. join собирает их с выбранной разделяющей строкой. Практика. Из фразы с повторными пробелами получить список слов и строку, в которой слова разделены запятыми.
	Итого	6	

Типовые ошибки и способы проверки.

– Программа обращается к первому символу пустой строки. Проверить длину до индексирования.
– Слова разделяются с явным пробелом и среди элементов появляются пустые. Для простого подсчёта применить split без аргумента.

Текущий контроль. Нормализовать введённую фразу, подсчитать слова и проверить наличие заданного отдельного слова.

Предъявляемые материалы. Код и три примера ввода, включая пустой.

Ожидаемый результат. Повторные пробелы не увеличивают счётчик, поиск отдельного слова не срабатывает на подстроку.

Модуль 5. Условия и выбор действий

Предметный результат. Строит взаимоисключающие ветви и объясняет, какой ввод к ним ведёт.

Содержание. Логическое выражение вычисляется в True или False. Сравнения объединяются с and, or и not. В цепочке if, elif, else выполняется первая подходящая ветвь. Вложенные проверки нужны, когда второй выбор зависит от первого. Уточняются границы диапазонов.

Практическая работа. Составить таблицу сценариев текстового приключения. Написать ветвления для выбора маршрута и проверку учебного правила допуска по двум условиям. Протестировать значения прямо на границе и рядом с ней.

Таблица 7 – Тематический план модуля 5

№	Тема	Часы	Содержание и учебное действие
1	Операции сравнения	1	Равенство проверяют через ==, присваивание записывают через =. Знаки < и <= различаются поведением на границе. Практика. Составить примеры для значений на единицу ниже, на пороге и выше порога.
2	Логические значения	1	True и False являются результатом проверки выражения. Их можно сохранить в переменную с именем, отражающим смысл вопроса. Практика. Сохранить результат сравнения очков с порогом и вывести его для двух наборов.
3	Два пути исполнения	1	if запускает блок при истинном условии, else обслуживает оставшиеся случаи. Отступы определяют состав блока. Практика. Сделать выбор одной из двух реплик и испытать оба пути.
4	Несколько вариантов	1	В цепочке if и elif выбирается первая истинная ветвь. Правила для смежных диапазонов удобнее сначала записать таблицей. Практика. Разделить результаты на три категории и подтвердить выходы на обеих границах.

Продолжение таблицы 7

№	Тема	Часы	Содержание и учебное действие
5	Сочетание условий	1	and требует выполнения обоих условий, or допускает выполнение хотя бы одного, not обращает результат. Скобки уточняют смысл сложного правила. Практика. Описать доступ при наличии билета и достаточного уровня. Проверить четыре комбинации.
6	Вложенная проверка	1	Вложенное условие вычисляется после попадания во внешний блок. Его применяют, когда второй вопрос имеет смысл только после первого ответа. Практика. В приключении сначала проверить выбранный путь, затем действие на нём. Проверить сценарии каждого пути.
	Итого	6	

Типовые ошибки и способы проверки.

– Равное порогу значение попало в неверную ветвь. Сверить в формулировке «больше» и «не меньше».
– Общее условие стоит перед частным и перехватывает его случаи. Изменить порядок либо сделать интервалы взаимоисключающими.

Текущий контроль. Реализовать проверку допуска по двум признакам и три результата для сценария приключения.

Предъявляемые материалы. Таблица решений, код и журнал прогонов по каждой ветке.

Ожидаемый результат. Граничные и недопустимые значения получают предусмотренный результат, выполняется одна подходящая ветвь.

Модуль 6. Цикл while и условие завершения

Предметный результат. Управляет повторением при неизвестном заранее количестве шагов.

Содержание. Цикл с предусловием проверяет необходимость очередного шага. Счётчик и накопитель имеют разные роли. Повторный запрос меняет состояние, от которого зависит завершение. Прерывание через break применяется при явно сформулированном событии.

Практическая работа. Сделать игру с фиксированным ответом и ограничением попыток. Реализовать меню с отдельной командой выхода. На бумаге проследить счётчик, затем воспроизвести и исправить цикл, который не меняет проверяемое значение.

Таблица 8 – Тематический план модуля 6

№	Тема	Часы	Содержание и учебное действие
1	Цикл с предусловием	1	while сначала проверяет условие и может не выполнить тело ни разу. Условие должно опираться на наблюдаемое состояние. Практика. Проследить два запуска цикла с начальными значениями, при которых он выполнится три раза и ни разу.
2	Обновление состояния	1	Переменная, от которой зависит условие, должна изменяться на нужных шагах. Отсутствие обновления может помешать завершению. Практика. Найти пропущенное обновление и показать исправленную трассировку значений.
3	Счётчик и накопитель	1	Счётчик показывает число событий, накопитель собирает сумму или другой итог. Начальные значения задаются до первого шага. Практика. Рассчитать суммарные очки за несколько попыток и отдельно вывести их количество.
4	Повторный запрос	1	Программа повторно предлагает ввод, пока пользователь не выполнит условие. Сообщение должно объяснять допустимое следующее действие. Практика. Запрашивать команду из короткого списка и проверить, что после неверной команды меню продолжает работу.
5	Выход через break	1	break немедленно завершает ближайший цикл. Условие выхода должно проверяться до действий, которые при завершении уже не нужны. Практика. Добавить команду выхода и подтвердить, что после неё не выводится лишний игровой запрос.
6	Безопасная остановка	1	Причина бесконечного цикла часто обнаруживается трассировкой условия и переменных. Для учебного прогона достаточно остановить выполнение средствами среды и исправить переход состояния. Практика. На подготовленном примере остановить зависший запуск, исправить условие и показать нормальное завершение.

Продолжение таблицы 8

№	Тема	Часы	Содержание и учебное действие
	Итого	6	

Типовые ошибки и способы проверки.

– Счётчик сбрасывается при каждом проходе. Инициализировать его до цикла, изменять внутри.
– Условие всегда истинно после нового ввода. Сверить, какую переменную проверяет while и какую обновляет тело.

Текущий контроль. Собрать игру с ограничением попыток и отдельное меню с командой выхода.

Предъявляемые материалы. Код и прогоны успешной игры, исчерпания попыток и выхода.

Ожидаемый результат. Циклы завершаются по заданным причинам, число попыток совпадает с правилом.

Модуль 7. Цикл for и перебор

Предметный результат. Выбирает перебор для заданной последовательности и соблюдает границы.

Содержание. Цикл for последовательно получает элементы строки или другой коллекции. range формирует последовательность чисел с исключённой верхней границей. По условию можно обновлять счётчик, а накопитель хранит промежуточный результат вычисления. Простые вложенные циклы демонстрируют перебор сочетаний.

Практическая работа. Посчитать в строке вхождения символа, сформировать таблицу вычислений и проверить число строк. Для короткого вложенного цикла предсказать пары значений до запуска.

Таблица 9 – Тематический план модуля 7

№	Тема	Часы	Содержание и учебное действие
1	Перебор элементов	1	for последовательно получает элементы коллекции. Для строки переменная цикла по очереди связывается с её символами. Практика. Вывести символы короткой строки по одному, предварительно указав число проходов.
2	Правая граница range	1	Верхняя граница диапазона с положительным шагом не включается. Для значений от 1 до 5 включительно диапазон должен иметь границу 6. Практика. Сформировать пять строк таблицы и проверить первую и последнюю.
3	Начало, конец и шаг	2	Три параметра range определяют начальное значение, границу и изменение. При отрицательном шаге диапазон идёт в сторону меньших чисел. Практика. Создать возрастающий ряд чётных чисел и обратный отсчёт с шагом 2.
4	Подсчёт по условию	2	Счётчик изменяется только при совпадении с критерием. Если совпадений нет, результат остаётся равным начальному нулю. Практика. Подсчитать букву «а» в двух строках, в том числе строке без неё.
5	Накопление суммы	2	Сумма после очередного прохода равна сумме предыдущих элементов плюс текущий. Промежуточные значения помогают проверить ход цикла. Практика. Сложить 2, 4 и 6, записав значения накопителя после каждого шага.
6	Вложенный перебор	2	Для каждого внешнего элемента внутренний цикл проходит свой диапазон. Число пар можно предварительно вывести из размеров двух последовательностей. Практика. Напечатать координаты сетки из двух строк и трёх столбцов и сверить число пар.
	Итого	10	

Типовые ошибки и способы проверки.

– В таблице пропало последнее нужное число. Сверить исключённую границу range.

– Сумма хранит только последнее значение. При каждом проходе прибавлять элемент к накопленной сумме.

Текущий контроль. Построить таблицу квадратов в заданном диапазоне и посчитать выбранные символы в коротком тексте.

Предъявляемые материалы. Код и проверка границ диапазона, пустого текста и повторов.

Ожидаемый результат. Обрабатываются ровно заданные значения и выводится ожидаемое число совпадений.

Модуль 8. Списки и изменение коллекций

Предметный результат. Хранит и изменяет коллекцию предсказуемым способом.

Содержание. Список хранит несколько значений и сохраняет их порядок. Индекс используется для чтения и замены, append добавляет элемент. Проверка длины предотвращает обращение к отсутствующей позиции. При присваивании нового имени тот же список остаётся общим. Перебор с одновременным удалением заменяется построением нового списка.

Практическая работа. Сделать список дел с добавлением, выводом и удалением позиции. На отдельном примере сравнить копирование и присваивание списка. Обработать пустой список и неверный индекс.

Таблица 10 – Тематический план модуля 8

№	Тема	Часы	Содержание и учебное действие
1	Создание списка и индекс	1	Список объединяет значения в определённой последовательности. Индекс выбирает позицию, а не ищет значение. Практика. Создать список трёх игр и вывести второй элемент.
2	Добавление элементов	1	append добавляет элемент в конец существующего списка. Метод изменяет коллекцию на месте. Практика. Добавить два балла по очереди и показать порядок вывода.
3	Изменение и удаление	1	Присваивание по индексу заменяет элемент, pop удаляет его по позиции и возвращает значение. Удаление сдвигает следующие элементы. Практика. Заменить среднюю задачу и удалить первую, сверив обе версии списка.

Продолжение таблицы 10

№	Тема	Часы	Содержание и учебное действие
4	Длина и пустой список	1	len даёт число элементов, которое определяет допустимые неотрицательные индексы. Пустой список требует отдельного сообщения при просмотре. Практика. Проверить меню просмотра без задач и после добавления одной записи.
5	Перебор с номерами	1	for передаёт сами элементы, а enumerate добавляет порядковый номер. Номер для отображения можно начать с 1. Практика. Вывести три задачи с номерами 1, 2, 3.
6	Копирование при изменениях	1	Присваивание одного списка другому имени не создаёт копию. При отборе безопасно перебирать исходный список и добавлять результаты в новый. Практика. Исключить два соседних элемента по условию, сохранив исходный список.
	Итого	6	

Типовые ошибки и способы проверки.

– После удаления соседние индексы сдвигаются. Пересчитать номер или отбирать элементы в новый список.
– Пункты меню нумеруются с 1, а индексы с 0. Преобразовать номер один раз после проверки.

Текущий контроль. Сделать меню добавления, просмотра и удаления задач.

Предъявляемые материалы. Код и протокол работы с пустым списком, тремя задачами и неверным номером.

Ожидаемый результат. Задачи выводятся по порядку, удаляется выбранный пункт, неверный номер сохраняет список.

Модуль 9. Поиск, подсчёт и сортировка

Предметный результат. Выбирает операцию обработки последовательности и проверяет граничные случаи.

Содержание. Последовательный поиск сравнивает элементы с запросом. Отбор строит новую коллекцию, счётчик накапливает число совпадений. Ми-

нимум и максимум требуют данных. sorted возвращает новый список, метод sort меняет существующий. Для одинаковых значений проверяется ожидаемое поведение.

Практическая работа. Обработать результаты игры. Показать исходный порядок, результат отбора по порогу и отдельный отсортированный вывод. Добавить пример с повторяющимися баллами и без результатов.

Таблица 11 – Тематический план модуля 9

№	Тема	Часы	Содержание и учебное действие
1	Наличие и поиск	1	Оператор in отвечает, есть ли значение в коллекции. Для нахождения всех позиций или записей по условию выполняется перебор. Практика. Сравнить поиск наличия балла с выводом всех его позиций в списке с повторами.
2	Минимум и максимум	1	min и max выбирают крайнее значение по общему порядку. Для отсутствия данных заранее определяется отдельный результат. Практика. Получить минимум и максимум для трёх наборов, включая пустой.
3	Подсчёт по условию	1	Счётчик увеличивается для каждого совпавшего элемента. Повторяющийся балл учитывается столько раз, сколько он записан. Практика. Посчитать результаты выше 10 в ряду 5, 10, 12, 12.
4	Построение отобранного списка	1	Новый список создаёт отдельный результат фильтрации. В него добавляют элементы, прошедшие одну и ту же проверку. Практика. Вывести отобранные баллы и показать неизменный исходный порядок.
5	Сортировка и обратный порядок	1	sorted формирует отсортированную копию, а reverse=True задаёт убывание. Обращение исходного порядка само по себе не сортирует данные. Практика. Сравнить убывающую сортировку и обращение последовательности 9, 2, 7.
6	Граничные случаи	1	Наборы из нуля или одного элемента помогают проверить предположения алгоритма. При повторяющихся значениях отдельно проверяются число записей и число уникальных баллов. Практика. Составить три граничных набора с ожидаемым результатом до запуска.

Продолжение таблицы 11

№	Тема	Часы	Содержание и учебное действие
	Итого	6	

Типовые ошибки и способы проверки.

– max вызван на пустом списке. До вычисления обработать отсутствие значений.
– Результат присваивания sort равен None. Использовать sorted или вызвать sort отдельно, если допустимо изменить список.

Текущий контроль. Отобрать результаты выше порога, посчитать их и показать лучшие баллы.

Предъявляемые материалы. Код, исходные данные и прогоны обычного, пустого и повторяющегося ввода.

Ожидаемый результат. Баллы на пороге исключаются, повторы учитываются, сортировка не меняет исходный порядок.

Модуль 10. Словари и множества

Предметный результат. Выбирает словарь для связей и множество для проверки уникальности.

Содержание. Словарь связывает уникальный ключ со значением. Проверка наличия ключа и получение значения с умолчанием позволяют обрабатывать отсутствующие записи. Словарь частот обновляется для каждого элемента. Множество хранит уникальные элементы и позволяет проверять принадлежность.

Практическая работа. Сделать справочник терминов с поиском и сообщение для отсутствующего ключа. Подсчитать частоты нормализованных слов, вывести уникальные слова и проверить повторяющуюся запись.

Таблица 12 – Тематический план модуля 10

№	Тема	Часы	Содержание и учебное действие
1	Ключ и значение	1	Ключ словаря однозначно определяет значение. Строковый ключ подходит для условного названия учебного термина. Практика. Создать три пары «термин и объяснение» и вывести значение одного ключа.
2	Получение и наличие ключа	1	Индексирование отсутствующего ключа вызывает ошибку. Проверка in или метод get помогает выбрать реакцию на отсутствие. Практика. Проверить два известных и один неизвестный термин.
3	Добавление и обновление	1	Присваивание по новому ключу добавляет запись. Повторное присваивание тому же ключу обновляет значение без изменения числа записей. Практика. Добавить термин, затем заменить его объяснение и сверить размер словаря.
4	Перебор пар	1	Метод items предоставляет ключ и значение каждой записи. Это позволяет сформировать отчёт с названиями и описаниями. Практика. Напечатать все строки справочника, отдельно показав число записей.
5	Уникальные элементы	1	Множество объединяет одинаковые элементы. Порядок для отчёта задают отдельно, когда он важен. Практика. Получить множество названий из списка с повтором и сверить число уникальных.
6	Таблица частот	1	Словарь частот сопоставляет токену число появлений. Новое слово начинается с нуля перед увеличением, существующее получает прежнее значение плюс один. Практика. Подсчитать частоты в строке «книга игра книга» и сверить сумму счётчиков с количеством слов.
	Итого	6	

Типовые ошибки и способы проверки.

– Неизвестный термин вызвал KeyError. Проверить наличие ключа или задать значение по умолчанию.
– Получилась отдельная запись для каждого регистра. Нормализовать токены по одинаковому правилу перед подсчётом.

Текущий контроль. Собрать справочник из нескольких записей и сформировать таблицу частот слов.

Предъявляемые материалы. Исходники, прогоны поиска известного и неизвестного термина и отчёт по повторяющимся словам.

Ожидаемый результат. Известный ключ даёт описание, неизвестный получает сообщение, повторы увеличивают частоту без дублирования ключей.

Модуль 11. Функции и декомпозиция

Предметный результат. Выделяет функции с понятными параметрами и результатом.

Содержание. Функция выделяет действие с определённым входом и результатом. Параметры получают значения аргументов. return передаёт результат вызывающему коду, print показывает его пользователю. Локальные переменные ограничены областью функции. Проверка одной функции упрощает поиск дефектов.

Практическая работа. Разделить викторину на показ вопроса, проверку ответа и подсчёт баллов. Написать отдельную функцию обработки текста, вызвать её с несколькими входами. Сравнить вывод на экран и возвращаемое значение.

Таблица 13 – Тематический план модуля 11

№	Тема	Часы	Содержание и учебное действие
1	Объявление и вызов	1	def создаёт функцию и определяет её тело. Само объявление не выполняет действия функции до вызова. Практика. Объявить функцию приветствия, вызвать дважды и предсказать число строк.
2	Передача аргументов	1	Параметры описывают, какие значения функция ожидает. Аргументы подставляются при конкретном вызове. Практика. Вызвать функцию форматирования с разными именами и темами.
3	Возврат результата	1	return завершает вызов и передаёт результат вызывающему коду. Результат можно сохранить или передать дальше. Практика. Создать функцию начисления одного балла за верный ответ и сложить результаты двух вызовов.

Продолжение таблицы 13

№	Тема	Часы	Содержание и учебное действие
4	Вывод и возврат	1	print отображает значение, а возвращаемое значение доступно другим частям программы. Для тестирования вычисления удобнее отделить от вывода. Практика. Сравнить две функции с одинаковым видимым сообщением и проверить их значения присваиванием.
5	Локальные имена	1	Имя, созданное внутри функции, принадлежит её локальной области. Разные вызовы получают собственные значения параметров. Практика. Показать, как одноимённая переменная вне функции сохраняется после вызова.
6	Декомпозиция	1	Отдельная функция полезна, если её задачу можно сформулировать и проверить независимо. Для викторины удобно отделить проверку ответа от показа вопросов. Практика. Выделить функции нормализации ответа и начисления баллов, проверить их по отдельности.
	Итого	6	

Типовые ошибки и способы проверки.

– Функция печатает результат, но вызывающий код получает None. Вернуть значение через return.
– Код работает лишь при определённом глобальном значении. Передать необходимое значение параметром.

Текущий контроль. Разделить получение вопроса, проверку ответа и подсчёт баллов на отдельные операции.

Предъявляемые материалы. Код и ожидаемые и фактические результаты отдельных вызовов.

Ожидаемый результат. Один и тот же набор правил работает для разных вопросов, ошибка в проверке выявляется отдельным запуском функции.

Модуль 12. Стандартные модули и случайные события

Предметный результат. Подключает стандартный модуль и объясняет используемые параметры.

Содержание. Импорт делает доступными готовые функции. Документация помогает уточнить параметры и результат. random выбирает элемент из непустой последовательности и формирует случайное целое в заданных границах. datetime представляет дату и время. Случайный сценарий удобнее проверять через предсказуемый учебный вход.

Практическая работа. Заменить фиксированный ответ игры случайным, сделать генератор заданий из списка и добавить отображение даты учебной записи. Вынести выбор ответа отдельно, чтобы проверять логику игры на известных значениях.

Таблица 14 – Тематический план модуля 12

№	Тема	Часы	Содержание и учебное действие
1	Импорт и имена модулей	1	import предоставляет имя модуля, через которое вызывают нужную функцию. Имя собственного файла не должно случайно перекрывать импортируемый стандартный модуль. Практика. Импортировать random и вызвать функцию через имя модуля.
2	Целое число из диапазона	1	randint выбирает целое число с включёнными границами. Границы должны быть согласованы с правилами игры. Практика. Проверить, что при одинаковых границах всегда получается заданное число.
3	Выбор из списка	1	choice возвращает один элемент непустой последовательности. Результат сравнивают с перечнем допустимых вариантов. Практика. Выбирать учебное задание из трёх строк и проверить принадлежность результата списку.
4	Перемешивание и повторяемость	1	shuffle изменяет список на месте. В контролируемом примере отдельный генератор с фиксированным seed позволяет повторить опыт в одинаковой среде. Практика. Дважды перемешать копии списка с одинаковой настройкой и сравнить порядок.

Продолжение таблицы 14

№	Тема	Часы	Содержание и учебное действие
5	Дата и время	1	Объекты datetime хранят поля даты и времени. Учебный отчёт может показывать дату отдельно от изменяемых игровых данных. Практика. Добавить к записи дату и проверить формат на заранее заданной календарной дате.
6	Чтение справки	1	Документация описывает параметры, возвращаемое значение и ограничения функции. Пример применения проверяют в небольшом самостоятельном фрагменте. Практика. Сравнить границы randint и range по справке и подтвердить отличие тестом.
	Итого	6	

Типовые ошибки и способы проверки.

– random.choice получает пустой список. Сначала проверить наличие доступных вариантов.
– После shuffle исчез исходный порядок. Перемешивать отдельную копию, если нужен первоначальный список.

Текущий контроль. Сформировать случайный выбор из списка и проверить механику игры на фиксированном ответе.

Предъявляемые материалы. Код, список допустимых значений и контрольный прогон с неизменным входом.

Ожидаемый результат. Случайный выбор не выходит за границы, логика игры проверяется без ожидания нужного случайного числа.

Модуль 13. Текстовые файлы и сохранение

Предметный результат. Читает и сохраняет текст с заданной кодировкой и режимом.

Содержание. Путь связывает программу с файлом в рабочей папке. Контекстный менеджер with закрывает файл после операций. Режим записи перезаписывает содержимое, режим добавления сохраняет прежние строки. UTF-8 задаёт совместимую кодировку для русского текста.

Практическая работа. Сохранить учебные заметки и результаты игры, прочитать их целиком и построчно. Перезапустить программу и сверить данные. На копии файла продемонстрировать разницу режимов записи.

Таблица 15 – Тематический план модуля 13

№	Тема	Часы	Содержание и учебное действие
1	Рабочая папка и путь	2	Относительный путь разрешается относительно текущей рабочей папки. Папка скрипта и рабочая папка могут различаться. Практика. Открыть учебный файл и объяснить, где он находится при выбранном способе запуска.
2	Контекст открытия файла	1	with управляет временем работы с открытым файлом. После выхода из блока файл закрывается даже при ошибке внутри. Практика. Показать чтение в блоке with и проверить результат после выхода.
3	Чтение целиком и построчно	1	read возвращает оставшийся текст целиком, а перебор файлового объекта выдаёт строки по очереди. Конец строки может включать символ перевода строки. Практика. Сравнить оба способа и вывести число записей.
4	Режимы записи	1	Режим w заменяет прежнее содержимое, а добавляет текст в конец. Для журнала необходима согласованная граница записи, например перевод строки. Практика. На двух копиях файла добавить новую строку разными режимами и сравнить содержание.
5	Кодировка текста	1	UTF-8 задаётся явно для одинакового чтения и записи русских строк. При несовпадающей кодировке содержимое может читаться с ошибкой. Практика. Записать слово с кириллицей и проверить его после чтения.
6	Перезапуск и сохранённое состояние	1	Переменные завершающегося процесса перестают хранить историю. Повторный запуск восстанавливает нужные данные из файла. Практика. Закрыть программу, запустить заново и показать первую запись до добавления второй.
	Итого	7	

Типовые ошибки и способы проверки.

– Файл создаётся в неожиданном месте. Показать текущую рабочую папку и проверить относительный путь.
– При каждой записи исчезает история. Сверить режимы w и a, испытать на копии данных.

Текущий контроль. Сохранять по одной строке результата и показывать накопленную историю при новом запуске.

Предъявляемые материалы. Код, исходный текстовый файл и два последовательных прогона.

Ожидаемый результат. Вторая запись добавлена без потери первой, русский текст читается корректно.

Модуль 14. Таблицы CSV и данные JSON

Предметный результат. Обрабатывает таблицу и структурированные настройки стандартной библиотекой.

Содержание. CSV представляет строки и поля таблицы. Готовый модуль учитывает разделители и кавычки. JSON поддерживает строки, числа, списки и словари. После загрузки проверяются ожидаемые поля и типы. Выбор формата зависит от структуры и назначения данных.

Практическая работа. Прочитать учебную таблицу результатов, вычислить сводку. Сохранить настройки в JSON, загрузить их и проверить поле с ожидаемым типом. Сопоставить исходные значения и прочитанную запись.

Таблица 16 – Тематический план модуля 14

№	Тема	Часы	Содержание и учебное действие
1	Строка данных и поле	2	Запись описывает один объект, а поле — конкретное свойство. Заголовок таблицы помогает согласовать имена и значения полей. Практика. Разметить три игровые записи и указать ожидаемый тип каждого поля.
2	Организация CSV	1	CSV хранит значения в строках с согласованными разделителями. Поле с запятой может быть заключено в кавычки, поэтому ручное разбиение ненадёжно. Практика. Прочитать две строки, одна из которых содержит название с запятой.

Продолжение таблицы 16

№	Тема	Часы	Содержание и учебное действие
3	Модуль csv	1	DictReader позволяет получать поля по заголовкам. Большинство значений приходит текстом, числовые поля преобразуются явно. Практика. Загрузить таблицу результатов и посчитать средний балл при наличии данных.
4	Структуры JSON	1	JSON кодирует массивы, объекты и простые значения. В Python массив обычно читается как список, а объект как словарь. Практика. Нарисовать структуру настроек с двумя полями и одним списком.
5	Загрузка и сохранение	1	Функции load и dump читают и записывают JSON через файловый объект. Сохранённое представление должно восстанавливаться в подходящую структуру. Практика. Записать настройки в UTF-8 и сверить их после повторной загрузки.
6	Проверка схемы	1	Синтаксически корректный JSON может содержать неподходящие поля. Программа отдельно проверяет ключи и допустимый тип значения. Практика. Подготовить три файла, обычный, с отсутствующим ключом и с числом вместо строки.
	Итого	7	

Типовые ошибки и способы проверки.

– Сравнение CSV-баллов идёт как сравнение строк. Преобразовать ожидаемые числовые поля после чтения.
– Программа принимает JSON с неожиданным типом значения. После загрузки проверить структуру и типы обязательных полей.

Текущий контроль. Подготовить сводку по CSV и сохранить параметры программы в JSON.

Предъявляемые материалы. Учебный CSV, файл настроек, исходники и прогоны с неверным полем.

Ожидаемый результат. Сводка соответствует входным строкам, настройки восстанавливаются, ошибка структуры объясняется пользователю.

Модуль 15. Диагностика ошибок и надёжный ввод

Предметный результат. Находит причину дефекта, исправляет её и проверяет результат.

Содержание. Синтаксическая ошибка возникает при разборе программы, исключение при её выполнении, а логический дефект обнаруживается сравнением ожидаемого и фактического результата. Сообщение помогает найти место сбоя. Обработчик try и except перехватывает конкретную ожидаемую ошибку.

Практическая работа. Воспроизвести три типа ошибок в готовом фрагменте. В калькуляторе повторять запрос после неверного числа. Добавить понятное сообщение при отсутствии файла и проверить, что другие ошибки остаются видны при отладке.

Таблица 17 – Тематический план модуля 15

№	Тема	Часы	Содержание и учебное действие
1	Синтаксический дефект	2	Интерпретатор сообщает о нарушении формы программы до выполнения проблемного файла. Указатель показывает место обнаружения, поэтому проверяется и соседняя строка. Практика. Исправить незакрытую скобку и подтвердить запуск.
2	Исключение во время работы	1	Исключение возникает при выполнении операции с неподходящим состоянием. Имя ошибки и цепочка вызовов помогают найти исходное выражение. Практика. Воспроизвести ValueError на неверном числе и найти проблемное преобразование.
3	Логический дефект	2	Программа с неверной формулой может завершаться без исключений. Для выявления нужен ожидаемый результат из условия или ручного расчёта. Практика. Найти неверный оператор в вычислении стоимости на двух проверочных примерах.
4	Промежуточные значения	2	Временный вывод или наблюдение в отладчике помогает локализовать расхождение. Проверка начинается с первого значения, которое отличается от ожидаемого. Практика. Проследить три промежуточных значения и назвать первую точку ошибки.

Продолжение таблицы 17

№	Тема	Часы	Содержание и учебное действие
5	Ожидаемое исключение	2	try ограничивает операции, в которых возможен известный сбой. except обрабатывает конкретное исключение и позволяет продолжить сценарий. Практика. Обработать неверное число, сохранив возможность повторного ввода.
6	Пустой ввод и отсутствующие файлы	2	Пустую строку удобно обнаружить до преобразования. Ошибка FileNotFoundError требует понятной подсказки или согласованного пустого состояния. Практика. Проверить пустую строку, текст вместо числа и отсутствующий файл отдельными прогонами.
	Итого	11	

Типовые ошибки и способы проверки.

– Общий обработчик скрывает дефект программы. Обрабатывать ожидаемый ValueError рядом с преобразованием.
– Исправление проверено только на ошибочном входе. Повторно прогнать рабочий пример и соседнюю границу.

Текущий контроль. Починить подготовленный калькулятор и добавить повторный запрос для неверного числа.

Предъявляемые материалы. Исправленный код и журнал дефектов с проверками.

Ожидаемый результат. Синтаксическая, исполнительная и логическая ошибки различены, корректный ввод даёт ожидаемый результат, неверный не завершает сценарий.

Модуль 16. Объекты и состояние

Предметный результат. Моделирует данные и действия через класс и экземпляры.

Содержание. Класс описывает общие свойства и действия. Экземпляр имеет собственное состояние. Начальные значения задаются при создании. Метод читает или меняет состояние, принимая необходимые параметры. Два экземпляра одного класса могут изменяться независимо.

Практическая работа. Создать класс игрового персонажа с именем и количеством очков. Добавить метод начисления баллов. Создать два экземпляра, обновить один и сравнить состояния до и после действия.

Таблица 18 – Тематический план модуля 16

№	Тема	Часы	Содержание и учебное действие
1	Объект и свойства	2	Объект объединяет связанные значения и доступные действия. Для персонажа имя и очки характеризуют его текущее состояние. Практика. Записать два экземпляра на карточках и отметить общие поля и разные значения.
2	Класс как описание	1	Класс задаёт общую форму объектов, но не хранит вместо каждого объекта его состояние. Один класс позволяет создавать несколько персонажей. Практика. Написать короткое объявление класса для учебной карточки.
3	Создание экземпляра	1	При создании экземпляра __init__ задаёт исходные атрибуты. Параметр self обозначает конкретный экземпляр внутри метода. Практика. Создать два объекта с разными именами и нулевым счётом.
4	Метод и действие	1	Метод вызывается через объект и может читать его атрибуты. Аргумент задаёт, например, сколько очков требуется начислить. Практика. Добавить метод начисления и показать значения до и после вызова.
5	Изменение состояния	1	Правило изменения состояния удобно проверять рядом с самим действием. Некорректный аргумент не должен оставлять объект частично изменённым. Практика. Установить правило неотрицательного начисления и проверить нормальное и отрицательное значения.
6	Независимость экземпляров	1	Атрибут, записанный через self в экземпляр, хранится отдельно от атрибутов других экземпляров. Атрибут класса доступен всем экземплярам, а изменение общего изменяемого объекта может повлиять на каждый из них. Тест изменения одного экземпляра выявляет случайно общее состояние. Практика. Изменить счёт одного персонажа и подтвердить неизменность второго.
	Итого	7	

Типовые ошибки и способы проверки.

– Очки обоих объектов меняются одновременно. Проверить, что состояние хранится в атрибуте экземпляра.
– Метод принимает отрицательные очки вопреки правилу. Проверить аргумент до изменения состояния.

Текущий контроль. Реализовать класс персонажа и управлять двумя независимыми экземплярами.

Предъявляемые материалы. Код и таблица состояний до и после начисления очков.

Ожидаемый результат. Действие меняет только выбранный экземпляр, а начальные значения каждого заданы явно.

Модуль 17. Рисование и события

Предметный результат. Строит рисунок из повторяемых действий и связывает событие с изменением сцены.

Содержание. Координаты задают положение в сцене. Циклы и функции помогают строить повторяющиеся фигуры. Событие вызывает обработчик, который обновляет состояние сцены. Видимый результат проверяется по последовательности действий пользователя и допустимым границам движения.

Практическая работа. Нарисовать орнамент в подготовленной графической среде. Реализовать перемещение фигуры по клавишам и ограничить выход за край. Проверить повторное событие и стартовое положение.

Таблица 19 – Тематический план модуля 17

№	Тема	Часы	Содержание и учебное действие
1	Координаты и перемещение	2	Точка сцены имеет координаты x и y. Изменение одной координаты перемещает объект вдоль выбранной оси. Практика. Поместить фигуру в три заданные точки и сверить ожидаемое направление.
2	Перо, цвет и рисунок	2	Направление, положение пера и цвет влияют на вид следа. Перемещение с поднятым пером не рисует линию. Практика. Нарисовать два разноцветных отрезка с промежутком.

Продолжение таблицы 19

№	Тема	Часы	Содержание и учебное действие
3	Повторяющийся узор	1	Цикл повторяет последовательность поворотов и движений. Геометрическое правило заранее определяет число повторов и общий угол. Практика. Нарисовать узор из одинаковых сегментов и сверить число повторов.
4	Функции для фигур	1	Функция выделяет способ построения одной фигуры, а её параметры описывают размеры или цвет. Это позволяет переиспользовать правило в разных частях сцены. Практика. Вызвать одну функцию рисования с двумя размерами.
5	События ввода	1	Событие запускает заранее зарегистрированную функцию, например при нажатии клавиши. Привязка сохраняет сам обработчик до появления события. Практика. Назначить две клавиши и проверить реакцию на каждую.
6	Состояние сцены	1	Координаты и другие свойства сцены обновляются после действий пользователя. Границы проверяются по предполагаемой новой позиции. Практика. Показать попытку выйти за край и повторное движение обратно.
	Итого	8	

Типовые ошибки и способы проверки.

– Нажатие клавиши не работает. Убедиться, что выбран нужный обработчик и окно принимает клавиатурный ввод.
– После движения остаются лишние линии. Разделить операции рисования и перемещения с поднятым пером.

Текущий контроль. Создать рисунок и управляемую фигуру, которая не выходит за границы сцены.

Предъявляемые материалы. Код, вид сцены и журнал трёх действий пользователя.

Ожидаемый результат. Видна повторяющаяся часть рисунка, обработчик перемещает фигуру, граничное движение ограничено.

Модуль 18. Диалоговый помощник

Предметный результат. Реализует команды и предсказуемые переходы диалога.

Содержание. Команда пользователя выбирает обработчик. Диалог использует данные и состояние для следующего шага. Команды помощи и сброса делают сценарий обратимым. Для подготовленных учащихся преподаватель показывает перенос команд в учебный бот и хранение токена вне исходного кода.

Практическая работа. Собрать локального помощника со справкой, несколькими командами и возвратом к началу. На контрольных сценариях проверить неизвестную команду и неверный ответ. Дополнительный вариант — подключить готовый учебный интерфейс мессенджера под контролем преподавателя. Локальный диалог является базовым вариантом. Интеграция с Telegram возможна при участии законного представителя и соблюдении условий сервиса.

Таблица 20 – Тематический план модуля 18

№	Тема	Часы	Содержание и учебное действие
1	Команда и обработчик	2	Обработчик объединяет действия, относящиеся к одной команде. Команду нормализуют перед поиском соответствия. Практика. Сопоставить три текстовые команды с обработчиками и проверить регистр и пробелы.
2	Ответы и подсказки	2	Ответ сообщает результат и возможный следующий шаг. Для справки полезен тот же список команд, который использует программа. Практика. Сформировать справку из двух команд и проверить соответствие их именам.
3	Данные диалога	1	Учебные вопросы и подсказки можно хранить отдельно от алгоритма. Словарь помогает получать запись по имени команды. Практика. Добавить справочную запись без изменения общей логики поиска.

Продолжение таблицы 20

№	Тема	Часы	Содержание и учебное действие
4	Состояние и возврат	1	Состояние обозначает текущий шаг и временные данные. Возврат к началу восстанавливает согласованные начальные значения. Практика. Пройти два шага, выполнить сброс и подтвердить отсутствие прежнего ответа.
5	Неизвестные команды	1	Нераспознанный ввод требует понятной реакции без изменения состояния. Повторный запрос позволяет восстановить сценарий. Практика. Показать восстановление после пустой и неизвестной команд.
6	Дополнительный перенос в бота	1	Интерфейс мессенджера передаёт вход существующей логике и возвращает ответ. Учебный токен хранится вне исходного кода, работа проводится в подготовленной среде преподавателя. Практика. Для базового маршрута показать локальный интерфейс; при дополнительном варианте проверить одну команду без включения токена в сдаваемые файлы.
	Итого	8	

Типовые ошибки и способы проверки.

– Неизвестная команда приводит к ошибке словаря. Проверить наличие обработчика и показать допустимые команды.
– После сброса сохранился предыдущий шаг. Очистить значения состояния вместе с переводом на стартовый экран.

Текущий контроль. Реализовать помощника со справкой, двумя командами и сбросом диалога.

Предъявляемые материалы. Исходники и протокол сценариев справки, ошибки и сброса.

Ожидаемый результат. Команды дают согласованные ответы, неизвестная команда не рушит программу, после сброса восстановлено начальное состояние.

Модуль 19. Планирование и разработка мини-проекта

Предметный результат. Реализует рабочую первую версию с выбранным способом хранения.

Содержание. Проект начинается с задачи пользователя и критериев готовности. Первая версия включает основное действие и минимальные данные. Декомпозиция связывает пользовательские сценарии с функциями. Малые изменения проверяются до добавления новых возможностей.

Практическая работа. Согласовать карточку проекта. Описать основной сценарий, выбрать структуры и формат хранения, собрать первую версию. Запустить её на двух наборах данных, зафиксировать замечания преподавателя и исправить блокирующий дефект.

Таблица 21 – Тематический план модуля 19

№	Тема	Часы	Содержание и учебное действие
1	Пользователь и первая версия	3	Первая версия решает конкретную задачу для выбранного пользователя. Ограничение объёма позволяет полностью проверить главный сценарий. Практика. Записать, кто будет пользоваться программой, и выбрать одно обязательное действие.
2	Сценарии использования	3	Сценарий связывает вход, шаги и ожидаемый итог. Дополнительные действия ранжируются по необходимости. Практика. Описать обычный и ошибочный сценарии и согласовать обязательный минимум.
3	Структура и хранение	2	Структура данных должна соответствовать поиску, изменению и сохранению записей. Формат выбирается по данным, а не только по внешнему виду файла. Практика. Сравнить текст, CSV и JSON для своего проекта и обосновать выбранный формат.
4	Разбиение кода	2	Функции отделяют ввод, обработку и сохранение. Разделение на файлы имеет смысл, когда это облегчает чтение и повторный запуск. Практика. Нарисовать цепочку вызовов и реализовать одну из функций отдельно.

Продолжение таблицы 21

№	Тема	Часы	Содержание и учебное действие
5	Пошаговая реализация	2	Каждый небольшой шаг должен завершаться работающей проверкой. Такой порядок помогает найти последнее изменение, повлиявшее на результат. Практика. Сначала реализовать ввод и вывод, затем хранение и подтвердить оба сценария.
6	Обратная связь и пересмотр	2	Замечания полезно связывать с наблюдаемым результатом и важностью для основной задачи. После исправления проверяется прежний и новый сценарий. Практика. Записать одно замечание преподавателя и показать, каким изменением оно закрыто.
	Итого	14	

Типовые ошибки и способы проверки.

– Список идей не позволяет понять, что считать готовой версией. Выбрать один пользовательский сценарий и критерий завершения.
– Изменены сразу несколько функций, и источник сбоя неясен. Разбить работу и проверять по одному шагу.

Текущий контроль. Согласовать мини-проект и собрать основное действие с сохранением данных.

Предъявляемые материалы. Карточка проекта, исходники и контрольный запуск.

Ожидаемый результат. Основной сценарий завершён, данные сохраняются, дальнейшие функции перечислены отдельно.

Модуль 20. Проверка, доработка и представление проекта

Предметный результат. Передаёт запускаемый проект с инструкцией и подтверждением проверок.

Содержание. Контрольный сценарий включает шаги, вход и ожидаемый результат. Повторная проверка после изменения помогает обнаружить возврат прежнего дефекта. Инструкция описывает среду, файлы и запуск. Презентация показывает результат и объясняет конкретное решение.

Практическая работа. Проверить основной сценарий, пустой или неверный ввод и повторное действие. Исправить найденный дефект, повторить про-

верки и подготовить инструкцию запуска. Провести пробный показ с коротким объяснением одной функции.

Таблица 22 – Тематический план модуля 20

№	Тема	Часы	Содержание и учебное действие
1	Сценарии проверки	3	Запись проверки включает вход, действие и ожидаемый результат. Набор должен покрывать ключевые возможности проекта. Практика. Составить минимум три проверки до демонстрации и проставить результаты.
2	Пустой и неверный ввод	3	Ошибочные данные и повторное действие помогают выявить слабые места. Реакция должна сохранять или предсказуемо обновлять состояние. Практика. Проверить пустой ввод, неверный тип и повтор главной команды.
3	Исправление и повторная проверка	4	Дефект сначала воспроизводят и описывают. После правки проверяют проблемный пример и соседние рабочие случаи. Практика. Зафиксировать один дефект до и после исправления с ожидаемым выводом.
4	Читаемость и сообщения	4	Понятные имена отражают назначение значений и функций. Сообщение пользователю сообщает, что произошло и как продолжить. Практика. Улучшить одно имя и одну подсказку, подтвердить сохранение поведения.
5	Инструкция запуска	4	Инструкция описывает среду, нужные файлы и порядок запуска. Проверку инструкции желательно поручить человеку, который не писал проект. Практика. Запустить проект по инструкции из отдельной копии и исправить пропущенный шаг.
6	Показ результата и развитие	4	На защите демонстрируют задачу, результат и одно существенное решение. Идея следующей версии опирается на наблюдаемое ограничение. Практика. Провести короткий показ, объяснить функцию и сформулировать одно проверяемое улучшение.
	Итого	22	

Типовые ошибки и способы проверки.

– Инструкция работает только в исходной рабочей папке. Проверить запуск из чистой копии со всеми перечисленными файлами.
– Исправление меняет результат соседнего сценария. Повторить весь короткий набор проверок после правки.

Текущий контроль. Передать работоспособный проект и провести короткую демонстрацию с разбором выбранной функции.

Предъявляемые материалы. Исходники, инструкция, три контрольных прогона и перечень исправлений.

Ожидаемый результат. Проект запускается по инструкции, обычный и ошибочный сценарии подтверждены, объяснение опирается на код.

2.4. Проектная практика и консультации

– Выбрать пользователя, цель и основной сценарий. Зафиксировать ожидаемую реакцию на пустой ввод.
– Определить структуру данных, формат хранения и минимум три проверки. Реализовать основное действие отдельно от диалога.
– Провести контрольный запуск с чистыми данными и повторный запуск с сохранёнными данными. Исправить один воспроизводимый дефект.
– Подготовить папку исходников, пример файла и короткую инструкцию с шагами запуска и ожидаемым результатом.

Консультации проводятся в составе практических часов соответствующих модулей. На консультации обучающийся предъявляет конкретный вопрос, выполненную попытку и обнаруженное расхождение. Педагог помогает выделить следующую проверку. Результат доработки обучающийся показывает самостоятельно.

3. Организационно-педагогические условия

3.1. Общие условия реализации

Обучение организуется как индивидуальная работа с преподавателем с демонстрацией, совместным разбором и самостоятельной попыткой. Перед усложнением педагог проверяет перенос базового действия на новый пример и

фиксирует затруднение в журнале. Для разных возрастов меняются объём опоры, размер примера и способ предъявления результата.

3.2. Кадровое и методическое обеспечение

Занятия ведёт педагог, соответствующий требованиям к должности и владеющий инструментами курса. Перед каждой новой темой педагог проверяет запуск учебного примера и готовит базовый и усложнённый варианты задачи. Во время занятия педагог сначала демонстрирует ключевое действие, затем наблюдает выполнение с опорой и предлагает короткий перенос на новый пример. В журнале фиксируются выполненная работа, степень самостоятельности и следующая цель.

3.3. Оборудование и учебная среда

- Компьютер с подключением к интернету, актуальный браузер, микрофон или гарнитура, возможность показать окно учебной среды.
- Python 3 и редактор или учебная IDE с консолью
- Стандартная библиотека Python, в том числе random, datetime, csv, json и turtle в предварительно проверенной графической среде с поддержкой клавиатурных событий
- Учебная платформа для сдачи работ и связи с преподавателем
- Подготовленные текстовые, CSV- и JSON-файлы без персональных данных

До старта преподаватель проверяет доступ к среде и запуск учебного файла. При техническом сбое занятие может быть продолжено на подготовленном материале с последующим самостоятельным выполнением ключевого действия под наблюдением преподавателя. Если проверяемую часть выполнить невозможно, её проводят в согласованное время.

3.4. Электронная информационно-образовательная среда

Используются личный кабинет ИННОПРОГ и средства видеосвязи, установленные приказом № ОБР-12. Доступ к заданиям, инструкциям, учебным файлам и обратной связи предоставляется на период обучения. Педагог иден-

тифицирует обучающегося по индивидуальной учётной записи и, если требуется, при видеосвязи. До зачисления семья получает сведения о технических требованиях, порядке доступа и действиях при сбое. Преподаватель фиксирует выполненные работы и результаты контроля. При недоступности платформы материалы передаются по согласованному каналу организации, а результаты затем переносятся в основной журнал. Перед допуском к работе проверяются права доступа и возможность открыть файл из учебной среды.

3.5. Здоровьесбережение, доступность и цифровая безопасность

Перед началом обучения педагог уточняет класс и вид устройства. Он планирует экранные фрагменты в пределах применимых норм и чередует их с устным разбором без экрана, бумажными заданиями, упражнениями для глаз и перерывами. Для обучающихся 1–4 классов использование ноутбука организуется с дополнительной клавиатурой в соответствии с применимыми требованиями. Длительность всей встречи включает виды работы без экрана. Продолжительность непрерывной работы с экраном, перерывы и организация рабочего места определяются с учётом возраста и применимых санитарных требований. При необходимости педагог предоставляет крупный текст, пошаговую инструкцию, дополнительные паузы, альтернативный способ ответа или время на освоение среды. Оценивание сохраняет предметные результаты программы.

Для учебных файлов используются обезличенные примеры. Работы передаются по каналам организации с доступом для участников обучения. Публикация материалов с персональными данными, ключами доступа или чужими ресурсами без разрешения исключается. Обучающийся указывает источник и условия использования заимствованных материалов.

4. Оценка качества освоения программы

4.1. Входная диагностика и текущий контроль

Входная диагностика проводится в начале первого раздела. Она помогает выбрать объём поддержки и носит диагностический характер. Для текущего контроля преподаватель применяет задания и признаки результата из рабочей программы, просит показать работу и объяснить хотя бы одно решение. При

ошибке обучающийся получает указание на конкретный проверяемый шаг и повторно предъявляет исправленный результат.

4.2. Практикумы и обратная связь

Практикумы включены в часы модулей «Цикл for и перебор» и «Диагностика ошибок и надёжный ввод». Педагог проверяет наблюдаемый результат работы, объясняет, что получилось, и предлагает конкретную доработку. После доработки обучающийся повторяет проверку. Задания и ожидаемые результаты приведены в приложении 2. Учебные проекты и демонстрация результата предусмотрены в рабочих программах модулей и приложении 3.

4.3. Внутренняя оценка качества реализации

По завершении цикла педагог сопоставляет долю выполненных контрольных работ, типовые ошибки и доступность учебной среды. Если одно затруднение систематически повторяется, пересматривается пример, порядок предъявления опор или формулировка задания. Изменение объёма и планируемых результатов оформляется в принятом организацией порядке.

5. Методические материалы

5.1. Указания для обучающихся

– Сохранять исходные данные, рабочий результат и пример проверки в отдельных файлах либо в разделах одного учебного артефакта.
– При затруднении описывать шаг, ожидаемый и фактический результат. Для исправленной ошибки показывать повторную проверку.
– При использовании подсказки или готового фрагмента указывать источник и уметь объяснить собственную часть работы.

5.2. Указания для преподавателя

– До занятия проверить пример и предусмотреть базовый вариант, дополнительное задание и контрольный случай.

– Различать ошибку понимания и ошибку инструмента. В первом случае попросить объяснить ход решения, во втором проверить среду и повторить действие.
– До практической работы сообщить ожидаемый результат и признаки выполнения. После показа сохранить работу, комментарий и следующий шаг.

5.3. Литература и электронные ресурсы

Страница курса ИННОПРОГ, сверка 8 ноября 2026 года. https://innoprog.ru/children/courses/python

– Python Tutorial. https://docs.python.org/3/tutorial/
– Python Standard Library. https://docs.python.org/3/library/

Приложение 1. Матрица формирования результатов

Таблица 23 – Связь результатов и контрольных материалов

Код	Планируемый результат	Модули	Свидетельство освоения
Р1	Создаёт и запускает файл Python, читает простой код и объясняет порядок выполнения команд.	1, 2	Код и вывод двух запусков с разными вымышленными данными.
Р2	Выбирает тип данных, формулирует условия, применяет циклы и отслеживает изменение значений на контрольных примерах.	2, 3, 5, 6, 7	Код и проверка границ диапазона, пустого текста и повторов.
Р3	Организует данные в списках, словарях и множествах, выполняет отбор, поиск, подсчёт и сортировку.	8, 9, 10	Исходники, прогоны поиска известного и неизвестного термина и отчёт по повторяющимся словам.
Р4	Разбивает решение на функции, применяет стандартные модули и различает локальные переменные, параметры и возвращаемые значения.	11, 12, 18	Исходники и протокол сценариев справки, ошибки и сброса.
Р5	Читает и сохраняет учебные данные в текстовом формате, CSV и JSON, проверяет структуру входных данных.	13, 14	Учебный CSV, файл настроек, исходники и прогоны с неверным полем.
Р6	Различает синтаксическую ошибку, исключение и логический дефект, воспроизводит сбой и проверяет исправление.	15, 19, 20	Исходники, инструкция, три контрольных прогона и перечень исправлений.
Р7	Моделирует состояние объекта или диалога и обновляет его по действиям пользователя.	16, 17, 18	Исходники и протокол сценариев справки, ошибки и сброса.
Р8	Представляет завершённый мини-проект, инструкцию запуска и протокол проверок.	19, 20	Исходники, инструкция, три контрольных прогона и перечень исправлений.

Приложение 2. Практикумы и обратная связь

Практикум «Цикл for и перебор»

Образец проверяемого действия. Построить таблицу квадратов в заданном диапазоне и посчитать выбранные символы в коротком тексте.

Материалы для проверки. Код и проверка границ диапазона, пустого текста и повторов.

Ожидаемый результат. Обрабатываются ровно заданные значения и выводится ожидаемое число совпадений.

Четыре академических часа практикума входят в часы модуля. Один час отводится на постановку задачи, два на работу и самопроверку, один на обсуждение и доработку.

Практикум «Диагностика ошибок и надёжный ввод»

Образец проверяемого действия. Починить подготовленный калькулятор и добавить повторный запрос для неверного числа.

Материалы для проверки. Исправленный код и журнал дефектов с проверками.

Ожидаемый результат. Синтаксическая, исполнительная и логическая ошибки различены, корректный ввод даёт ожидаемый результат, неверный не завершает сценарий.

Четыре академических часа практикума входят в часы модуля. Один час отводится на постановку задачи, два на работу и самопроверку, один на обсуждение и доработку.

Преподаватель записывает, что ученик сделал самостоятельно, где потребовалась подсказка и какой результат предстоит доработать. Ученик показывает исправление на том же или новом контрольном примере.

Приложение 3. Учебные проекты

Представить самостоятельный мини-проект на Python. Показать основной пользовательский сценарий и сохранение данных, воспроизвести обработку неверного ввода и объяснить решение по одной функции или структуре данных.

Материалы для обратной связи. Исходный код, учебные данные или пример файла, инструкция запуска, протокол минимум трёх проверок и демонстрация.

Варианты задания при сопоставимой сложности.

– Консольная викторина с подсчётом баллов и журналом результатов.
– Диалоговый помощник с каталогом команд и сохранением настроек.
– Каталог учебной коллекции с поиском и загрузкой из JSON или CSV.

Признаки освоения.

– Проект запускается по инструкции и выполняет согласованный основной сценарий.
– Данные представлены подходящей структурой и сохраняются в выбранном формате.
– Реализована проверка ожидаемого неверного ввода, а контрольный пример воспроизводим.
– Обучающийся объясняет назначение части кода и подтверждает результат проверкой.
– Материалы не содержат реальных персональных данных и ключей доступа.

Пример учебной работы

Создайте консольную викторину из трёх вопросов. Вопросы и ответы хранятся в структуре данных. Программа принимает номер варианта ответа, отклоняет пустой и неизвестный ввод и сохраняет результат прохождения в учебный файл. Вводите вымышленные имена либо используйте номер попытки.

Проверка. На первой попытке дайте три правильных ответа и получите 3 балла. На второй введите пустую строку и недопустимый номер, затем ответь-

те на один вопрос правильно. Проверьте, что ошибки ввода не меняют счёт, а после повторного запуска в файле остаются обе завершённые попытки.

Лист обратной связи по работе

Таблица 24 – Лист обратной связи

Поле	**Запись**
Обучающийся и дата	
Маршрут или вариант	
Материалы работы	
Проверочные сценарии	
Наблюдения по критериям	
Результат и следующий шаг	
Замечания и срок доработки	
Преподаватель	

Приложение 4. Памятка по подготовке работы

- Выбрать пользователя, цель и основной сценарий. Зафиксировать ожидаемую реакцию на пустой ввод.
- Определить структуру данных, формат хранения и минимум три проверки. Реализовать основное действие отдельно от диалога.
- Провести контрольный запуск с чистыми данными и повторный запуск с сохранёнными данными. Исправить один воспроизводимый дефект.
- Подготовить папку исходников, пример файла и короткую инструкцию с шагами запуска и ожидаемым результатом.

В итоговую папку или карточку работы включаются только материалы, необходимые для воспроизведения результата. Имя файла, порядок запуска, учебные данные и ожидаемый результат указываются явно.

Приложение 5. Лист актуализации программы

Лист заполняется после утверждения изменений. При актуализации сверяются возраст, общий объём, календарный график, результаты, материалы контроля и согласованность с публичным описанием курса.

Таблица 25 – Регистрация изменений программы

Редакция	Дата	Описание изменения	Основание