ИНДИВИДУАЛЬНЫЙ ПРЕДПРИНИМАТЕЛЬ
СУББОТИН ВЛАДИМИР АЛЕКСАНДРОВИЧ

ИНН 440102468242

УТВЕРЖДАЮ
Индивидуальный предприниматель
_____ / В. А. Субботин
«21» октября 2026 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

технической направленности

«Основы программирования на Python. Разработка текстовой ролевой игры»

Уровень программы	ознакомительный (стартовый)
Возраст обучающихся	7–17 лет
Объём программы	48 академических часов
Срок реализации	48 занятий; срок по индивидуальному графику
Форма обучения	очная
Технология реализации	исключительно электронное обучение и дистанционные образовательные технологии

Автор-составитель: Субботин Владимир Александрович
Кострома • 2026

Содержание

1. Комплекс основных характеристик программы

2. Комплекс организационно-педагогических условий

3. Учебный план

4. Содержание учебного плана

5. Календарный учебный график

6. Формы контроля и оценочные материалы

7. Условия реализации программы и электронная среда

8. Методические материалы

9. Воспитательный компонент

10. Список нормативных и учебных источников

Приложение 1. Карта оценки итогового проекта

Нормативная основа

Программа разработана с учётом Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», приказа Минпросвещения России от 27.07.2022 № 629 и Правил применения электронного обучения и дистанционных образовательных технологий, утверждённых постановлением Правительства РФ от 11.10.2023 № 1678 (в действующей редакции).

1. Комплекс основных характеристик программы

1.1. Пояснительная записка

Направленность программы: техническая.

Уровень освоения: ознакомительный (стартовый).

Язык реализации: русский.

Программа знакомит обучающихся с базовыми конструкциями языка Python через последовательную разработку текстовой ролевой игры. На каждом этапе ученик добавляет в единый проект новый элемент: героя, характеристики, диалоги, развилки сюжета, игровой цикл, случайные события, сражения, инвентарь, локации и сохранение состояния. Такой формат связывает каждую тему программирования с понятным результатом и формирует первый опыт самостоятельной разработки игры.

1.2. Актуальность

Программирование развивает алгоритмическое мышление, способность декомпозировать задачи, проверять гипотезы и исправлять ошибки. Python подходит для первого знакомства с кодом благодаря читаемому синтаксису и возможности быстро получать видимый результат. Программа переводит интерес детей к цифровым продуктам и играм из потребительской позиции в созидательную: обучающийся проектирует правила, пишет код, тестирует и представляет собственное приложение.

1.3. Отличительные особенности

- единая проектная линия: от первых команд до законченной текстовой ролевой игры;
- индивидуальная практическая работа на каждом занятии без обязательной командной разработки;
- постепенное усложнение: герой и данные → выборы и условия → игровой цикл → инвентарь и локации → функции и модули → обработка ошибок, классы и JSON → финальная сборка и защита;
- обучение отладке как отдельному навыку: чтение сообщений об ошибках, поиск причины, проверка исправления;
- полностью дистанционный формат с фиксацией работ, обратной связи и результатов в электронной информационно-образовательной среде;
- возможность получения подсказок от встроенного ИИ-помощника при сохранении самостоятельности выполнения контрольных и проектных работ.

1.4. Адресат программы

Программа предназначена для обучающихся 7–17 лет, интересующихся компьютерами, играми и созданием цифровых продуктов. Предварительный опыт программирования не требуется. Для зачисления не проводятся конкурсный отбор и вступительные испытания. Необходимы базовые навыки работы с компьютером: запуск программ, ввод текста, сохранение и открытие файлов.

Формат обучения: индивидуальные занятия с одним обучающимся. Формирование учебных групп программой не предусмотрено. Содержание объяснений, темп работы и объём учебной поддержки адаптируются к возрасту, уровню подготовки и индивидуальным особенностям обучающегося без изменения общего объёма программы и планируемых результатов.

При необходимости индивидуальный предприниматель определяет специальные условия освоения программы обучающимся с ограниченными возможностями здоровья на основании заявления и имеющихся рекомендаций.

1.5. Объём, срок и режим занятий

Параметр	Значение
Объём	48 академических часов
Срок	48 занятий; при одном занятии в неделю — 48 учебных недель
Режим	по индивидуальному расписанию; продолжительность каждой встречи — 60 минут
Учебная нагрузка	45 минут учебной работы (1 академический час), 10 минут перерыва и 5 минут технического подключения
Основной режим	синхронные онлайн-занятия с педагогом; материалы и задания доступны в ЭИОС
Самостоятельная работа	рекомендуется дополнительно, в объём 48 часов не включается

Встреча длится ровно 60 минут. В учебном плане учитываются 45 минут учебной работы; перерыв и техническое подключение в учебные часы не включаются. Программа

включает 48 таких встреч (48 академических часов, или 36 астрономических часов учебной работы). Работа с экраном ПК или ноутбука, включая вход, просмотр демонстрации, написание кода и сохранение файлов, суммарно за встречу не превышает: 1–2 классы — 20 минут; 3–4 классы — 25 минут; 5–9 классы — 30 минут; 10–11 классы и 1–2 курсы СПО — 35 минут. Для остальных обучающихся 16–17 лет программой также принят предел 35 минут. Вне экранных блоков ученик работает на бумаге и общается голосом, не глядя на экран. Педагог учитывает предшествующую дневную нагрузку и при необходимости сокращает экранные блоки. Режим закрепляется локальным актом ИП.

1.6. Форма обучения и особенности дистанционной реализации

Форма обучения — очная. Программа реализуется исключительно с применением электронного обучения и дистанционных образовательных технологий. Все учебные занятия, текущий контроль и итоговый контроль проводятся удалённо через электронную информационно-образовательную среду и средства видеоконференцсвязи. Очное присутствие обучающихся по адресу индивидуального предпринимателя не предусматривается.

Взаимодействие проводится в режиме реального времени (онлайн-занятие, демонстрация экрана, устный опрос, защита проекта) и при необходимости в отложенном режиме (учебные материалы, задания, комментарии педагога, загрузка файлов).

1.7. Цель и задачи

Цель: сформировать у обучающихся основы алгоритмического и проектного мышления и первоначальные навыки разработки текстовой ролевой игры на Python.

Обучающие задачи:

- познакомить с синтаксисом Python и средой разработки;
- научить использовать переменные, типы данных, ввод и вывод, условия, циклы, списки, словари, функции, обработку исключений, классы и работу с JSON;
- сформировать навыки проектирования сюжета, игровых состояний, диалогов, системы выбора, инвентаря, сражений и локаций;
- научить собирать изученные конструкции в единый проект текстовой ролевой игры;
- научить читать сообщения об ошибках, тестировать и улучшать код.

Развивающие задачи:

- развивать логическое, алгоритмическое и системное мышление;
- развивать самостоятельность в постановке и решении учебной задачи;
- формировать навыки планирования, самопроверки и презентации результата.

Воспитательные задачи:

- формировать ответственное и безопасное поведение в цифровой среде;
- воспитывать уважение к интеллектуальному труду и авторским правам;
- формировать настойчивость, аккуратность и конструктивное отношение к ошибке.

1.8. Планируемые результаты

По завершении программы обучающийся:

Предметные результаты:

- создаёт, сохраняет и запускает программы Python в среде разработки;
- объясняет назначение переменных и основных типов данных;
- использует ввод, вывод, арифметические операции и f-строки;

- строит алгоритмы с ветвлениями и циклами;
- применяет списки, словари, пользовательские функции, обработку исключений и базовые классы;
- проектирует ветвящийся сюжет, игровые состояния, инвентарь, локации и простую систему сражений;
- собирает отдельные механики в работоспособную текстовую ролевую игру, сохраняет состояние в JSON и улучшает архитектуру проекта;
- находит типовые синтаксические и логические ошибки;
- разрабатывает и дистанционно защищает индивидуальный итоговый проект.

Метапредметные результаты:

- разбивает задачу на последовательность действий;
- сравнивает ожидаемый и фактический результат;
- использует обратную связь для улучшения работы;
- кратко объясняет замысел, устройство и результат проекта.

Личностные результаты:

- проявляет самостоятельность и ответственность за цифровой результат;
- соблюдает правила безопасной и этичной работы с информацией;
- воспринимает ошибку как источник информации для улучшения решения.

2. Комплекс организационно-педагогических условий

2.1. Организация обучения

Зачисление несовершеннолетнего осуществляется на основании заявления родителя (законного представителя) и договора об образовании; совершеннолетний обучающийся подаёт заявление и заключает договор самостоятельно. До начала обучения обучающемуся, а для несовершеннолетнего также его родителю, предоставляются сведения о программе, расписании, технических требованиях, правилах работы в электронной среде, способах связи и порядке обработки персональных данных.

Обучение ведётся только индивидуально, в формате «один педагог — один обучающийся». Каждое занятие включает постановку задачи, короткое объяснение, демонстрацию, самостоятельную практику, проверку результата и рефлексию. Педагог вправе адаптировать темп и способы объяснения к возрасту обучающегося и изменять соотношение видов деятельности внутри темы без изменения общего объёма программы и планируемых результатов.

Сведения об индивидуальном предпринимателе: ИП Субботин Владимир Александрович, ИНН 440102468242. Адрес: 156539, Россия, Костромская область, Костромской, Кузьминка, ул. Деревенская, 8А. Приём обучающихся и проведение очных занятий по этому адресу программой не предусмотрены.

2.2. Условия проведения занятий

Занятия проводятся индивидуально в дистанционном формате. Для реализации программы используются Python 3, Visual Studio Code, Контур.Толк и электронная среда, обеспечивающая доступ к материалам, приём работ, обратную связь и фиксацию результатов.

Содержание занятий и способы объяснения адаптируются к возрасту и исходной подготовке обучающегося. На занятиях чередуются работа за компьютером и упражнения без экрана.

2.3. Формы занятий

- онлайн-урок с объяснением и демонстрацией экрана;
- практикум с индивидуальным выполнением задания;
- разбор ошибок и рефакторинг кода;
- мини-проект по завершении модуля;
- индивидуальная консультация;
- дистанционная защита итогового проекта.

2.4. Документирование результатов

Ход освоения программы фиксируется в электронной форме: посещаемость онлайн-занятий, выполнение заданий, комментарии педагога, результаты текущего и итогового контроля, а также работы обучающегося. Сроки и порядок хранения определяются локальными актами индивидуального предпринимателя с соблюдением законодательства о персональных данных и архивном деле.

2.5. Документ об обучении

Обучающемуся, успешно освоившему программу, выдаётся сертификат об обучении по образцу и в порядке, утверждённым ИП Субботиным В. А. Сертификат не удостоверяет получение уровня образования или профессиональной квалификации.

3. Учебный план

№	Раздел / тема	Всего	Теория	Практика	Контроль
1	Модуль 1. Начало текстовой RPG	6	2	4	мини-проект
1.1	VS Code. Первая сцена	1	0,5	0,5	наблюдение
1.2	Текст, print и оформление истории	1	0,5	0,5	практическая задача
1.3	input, диалоги и имя героя	2	0,5	1,5	практическая задача
1.4	Практика: пролог игры	2	0,5	1,5	защита мини-проекта
2	Модуль 2. Герой и игровые параметры	5	1,5	3,5	мини-проект
2.1	Переменные, типы и характеристики героя	2	0,5	1,5	практическая задача
2.2	Здоровье, опыт и игровые ресурсы	1	0,5	0,5	практическая задача
2.3	Практика: карточка героя	2	0,5	1,5	защита мини-проекта
3	Модуль 3. Выборы и ветвящийся сюжет	6	2	4	мини-проект
3.1	Условия if, elif, else	2	1	1	практическая задача
3.2	Вложенные условия и последствия выбора	2	0,5	1,5	практическая задача
3.3	Практика: сюжетная развилка	2	0,5	1,5	защита мини-проекта
4	Модуль 4. Игровой цикл и сражения	5	1,5	3,5	мини-проект
4.1	Цикл while и основной игровой цикл	2	0,5	1,5	практическая задача
4.2	Цикл for и случайные события	1	0,5	0,5	практическая задача
4.3	Практика: система сражения	2	0,5	1,5	защита мини-проекта
5	Модуль 5. Инвентарь, персонажи и локации	6	2	4	мини-проект
5.1	Списки и инвентарь	2	1	1	практическая задача
5.2	Словари: герой, противники и предметы	2	0,5	1,5	практическая задача
5.3	Практика: система локаций	2	0,5	1,5	защита мини-проекта
6	Модуль 6. Сборка версии 1.0 текстовой RPG	8	2,5	5,5	промежуточный проект
6.1	Функции и игровые действия	2	1	1	практическая задача
6.2	Модули и сохранение состояния	2	0,5	1,5	практическая задача
6.3	Сборка и расширение игры	2	0,5	1,5	проверка этапа

№	Раздел / тема	Всего	Теория	Практика	Контроль
6.4	Тестирование и защита версии 1.0	2	0,5	1,5	защита версии проекта
7	Модуль 7. Расширение и финализация текстовой RPG	12	3,5	8,5	итоговый проект
7.1	Надёжный ввод и обработка ошибок	2	0,5	1,5	практическая задача
7.2	Классы: герой и противник	2	1	1	практическая задача
7.3	JSON: сохранение и загрузка	2	0,5	1,5	практическая задача
7.4	Квесты и генерация событий	2	0,5	1,5	практическая задача
7.5	Рефакторинг, баланс и тестирование	2	0,5	1,5	проверка этапа
7.6	Финальная сборка и дистанционная защита	2	0,5	1,5	защита проекта
	ИТОГО	48	15	33	

4. Содержание учебного плана

4.1. Модуль 1. Начало текстовой RPG

Теория. Назначение Python и интерфейс Visual Studio Code. Файлы .py, запуск программы, функция print(), строки, комментарии, input() и f-строки. Структура игровой сцены: описание, реплика, действие и ответ игрока.

Практика. Настройка проекта в VS Code; запуск первой программы; оформление текста; запрос имени героя; создание диалога; разработка интерактивного пролога текстовой RPG.

4.2. Модуль 2. Герой и игровые параметры

Теория. Переменные и типы str, int, float, bool. Арифметические операции и преобразование типов. Игровое состояние: здоровье, опыт, золото, сила и другие характеристики персонажа.

Практика. Создание карточки героя; изменение характеристик после событий; подсчёт урона, лечения и награды; вывод текущего состояния персонажа.

4.3. Модуль 3. Выборы и ветвящийся сюжет

Теория. Логические значения, сравнения, if, elif, else и вложенные условия. Представление сюжетной развилки и последствий решения игрока.

Практика. Проверка выбора пользователя; сцены с несколькими путями; доступ к событию по характеристике героя; создание законченной сюжетной развилки с разными исходами.

4.4. Модуль 4. Игровой цикл и сражения

Теория. Циклы while и for, range(), break и модуль random. Основной игровой цикл, повторение хода, случайные события и условие завершения игры.

Практика. Цикл выбора действия; бросок случайного события; пошаговый обмен ударами; изменение здоровья героя и противника; победа, поражение и выход из сражения.

4.5. Модуль 5. Инвентарь, персонажи и локации

Теория. Списки и словари: индексация, append(), pop(), remove(), len(), перебор, ключи и значения. Представление предметов, персонажей, противников и связанных локаций.

Практика. Добавление и использование предметов; описание героя и противников словарями; переходы между локациями; проверка доступности пути и создание небольшой карты мира.

4.6. Модуль 6. Сборка версии 1.0 текстовой RPG

Теория. Функции, параметры, return, области видимости, модули и принцип DRY. Простое сохранение состояния в файл. Планирование минимальной версии игры и критерии готовности.

Практика. Перенос игровых механик в функции; разделение проекта на файлы; сохранение и загрузка состояния; объединение сюжета, локаций, инвентаря и сражений; тестирование и исправление ошибок; демонстрация работоспособной версии 1.0.

4.7. Модуль 7. Расширение и финализация текстовой RPG

Теория. Надёжный ввод и обработка исключений try/except; базовые принципы объектно-ориентированного программирования, классы и объекты; формат JSON; разделение данных и логики; принципы рефакторинга, балансировки и тестирования игровых сценариев.

Практика. Добавление обработки ошибочного ввода; создание классов героя и противника; сохранение и загрузка состояния в JSON; расширение системы квестов и случайных событий; балансировка механик; рефакторинг и тестирование; финальная сборка и дистанционная защита расширенной версии игры.

5. Календарный учебный график

Конкретные даты начала и окончания обучения, дни и время занятий определяются индивидуальным расписанием, оформляемым до начала обучения. Ниже указана последовательность модулей. При одном занятии в неделю курс длится 48 учебных недель; при другом индивидуальном расписании срок определяется датами занятий. Одно занятие включает 45 минут учебной работы, 10 минут перерыва и 5 минут подключения. В учебном плане учитываются 48 академических часов.

Занятия	Раздел	Часы	Контроль	Недели при 1 занятии в неделю
1–6	Начало текстовой RPG	6	мини-проект	1–6
7–11	Герой и игровые параметры	5	мини-проект	7–11
12–17	Выборы и ветвящийся сюжет	6	мини-проект	12–17
18–22	Игровой цикл и сражения	5	мини-проект	18–22
23–28	Инвентарь, персонажи и локации	6	мини-проект	23–28
29–36	Сборка версии 1.0 текстовой RPG	8	промежуточный проект	29–36
37–48	Расширение и финализация текстовой RPG	12	защита проекта	37–48

6. Формы контроля и оценочные материалы

6.1. Входная диагностика

Проводится на первом занятии в форме короткой беседы и практических действий: открыть программу, набрать и запустить код, сохранить файл. Результат не влияет на освоение программы и используется педагогом для выбора темпа и объёма поддержки.

6.2. Текущий контроль

- устный опрос и объяснение обучающимся логики своего решения;

- наблюдение за выполнением практической работы при демонстрации экрана;
- автоматическая или ручная проверка кода и результата запуска;
- мини-проект по завершении каждого модуля;
- самопроверка по чек-листу и краткая рефлексия.

6.3. Итоговый контроль

Форма итогового контроля — дистанционная защита индивидуального программного проекта в режиме видеоконференцсвязи. Обучающийся демонстрирует экран, запускает приложение, объясняет ключевые участки кода и отвечает на вопросы педагога. Идентификация проводится путём входа в персональную учётную запись и визуального подтверждения личности педагогом; иной способ может быть установлен локальным актом индивидуального предпринимателя.

Для получения зачёта необходимо: представить работоспособный проект; набрать не менее 6 из 10 баллов по карте оценивания; продемонстрировать понимание основных элементов собственного кода. При техническом сбое защита переносится на согласованное время без снижения результата.

Интерпретация результата: 9–10 баллов — высокий уровень; 6–8 баллов — достаточный уровень; 0–5 баллов — требуется доработка проекта и повторная защита.

6.4. Примерные задания текущего контроля

1. Запросить имя героя и вывести оформленное вступление к истории.
2. Изменить здоровье и количество золота героя после игрового события.
3. Создать сюжетный выбор с двумя вариантами и разными последствиями.
4. Создать игровой цикл, который повторяет выбор действия до команды выхода.
5. Добавить предмет в инвентарь и применить его к герою.
6. Описать функцию атаки и проверить её на нескольких противниках.
7. Добавить обработку некорректного ввода с помощью try/except.
8. Создать класс героя или противника и использовать его в игровом сценарии.
9. Сохранить состояние героя в JSON-файл и восстановить его после перезапуска.
10. Провести рефакторинг одного игрового модуля и проверить не менее трёх сценариев.

7. Условия реализации программы и электронная среда

7.1. Электронная информационно-образовательная среда

Для дистанционного освоения программы используются электронные учебные материалы, средства видеосвязи, приёма работ, обратной связи и фиксации результатов обучения. Используемые сервисы и инструкции по доступу доводятся до обучающегося и его родителя до начала занятий.

7.2. Используемые средства

Учебные материалы размещаются на Neurotoads (https://neurotoads.ru/); синхронные занятия проходят в Контур.Толк. Для программирования используются Python 3 и Visual

Studio Code. Сервис для сдачи работ, обратной связи и учёта результатов определяется утверждённым ИП перечнем и сообщается обучающемуся до начала занятий.

7.3. Технические требования к обучающемуся

- персональный компьютер или ноутбук с исправной клавиатурой;
- стабильное подключение к сети Интернет;
- актуальный браузер;
- микрофон и, при необходимости идентификации, веб-камера;
- поддерживаемая версия Python 3, совместимая с учебными материалами; Visual Studio Code (VS Code) и расширение Python от Microsoft; интерпретатор Python устанавливается отдельно;
- возможность сохранять и загружать файлы формата .py.

7.4. Идентификация, контроль и технические сбои

Доступ к учебным материалам и сдаче работ предоставляется через индивидуальную учётную запись в утверждённой ЭИОС. Передача данных для входа третьим лицам не допускается. При контроле педагог сопоставляет данные обучающегося с участником занятия. Способ идентификации и действия при сомнениях закрепляются локальным актом до начала обучения.

При техническом сбое обучающийся или родитель сообщает педагогу по резервному каналу, доведённому до них заранее и не зависящему от доступности сайта. Педагог фиксирует сбой, помогает восстановить подключение или переносит контроль. Сбой сам по себе не снижает оценку. Контакты и порядок действий указываются в договоре либо памятке до начала обучения.

7.5. Применение ИИ-помощника

При доступности ИИ-помощник помогает разбирать задания и ошибки. Во время контроля обучающийся самостоятельно объясняет код и решение; при отсутствии помощника поддержку обеспечивает педагог.

8. Методические материалы

8.1. Методические принципы

- от простого к сложному: новая конструкция опирается на уже созданный код;
- короткое объяснение — немедленная практика — проверка результата;
- минимально достаточная подсказка вместо выдачи готового решения;
- видимый результат каждого занятия;
- индивидуальный темп в пределах планируемых результатов;
- обязательное проговаривание обучающимся логики собственного решения.

8.2. Организация онлайн-занятия

Занятие включает 45 минут учебной работы, 10 минут перерыва и 5 минут технического подключения. Педагог чередует работу за компьютером и задания без экрана с учётом возраста обучающегося и ограничений, указанных в разделе 1.5.

8.3. Методическое обеспечение

Для проведения занятий используются учебные материалы по темам, практические задания, демонстрационные файлы Python, инструкции по установке программ и критерии оценки итогового проекта.

9. Воспитательный компонент

Воспитательная работа интегрирована в содержание занятий и направлена на формирование ответственного отношения к цифровой деятельности, уважения к интеллектуальному труду, самостоятельности и культуры конструктивной обратной связи.

10. Список нормативных и учебных источников

10.1. Нормативные источники

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (в действующей редакции).
2. Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Постановление Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ» (в действующей редакции).
4. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» (в действующей редакции).

10.2. Учебные и электронные ресурсы

1. Python Software Foundation. Python 3 Documentation: https://docs.python.org/3/
2. Microsoft. Python в Visual Studio Code: https://code.visualstudio.com/docs/python/python-tutorial
3. Электронные материалы курса на платформе Neurotoads: https://neurotoads.ru/.

Приложение 1. Карта оценки итогового проекта

Критерий	0 баллов	1 балл	2 балла
Работоспособность	не запускается или основная функция не работает	частично работает; есть существенные ошибки	работает устойчиво по основным сценариям
Изученные конструкции	не применены или применены случайно	применены с помощью педагога	применены уместно и объяснены
Качество кода	структура затрудняет чтение	в целом понятен, есть повторения	понятные имена и логичная структура
Тестирование	проверка не проводилась	проверен один сценарий	проверены разные сценарии и исправлены ошибки
Защита	не может объяснить решение	объясняет часть решения с подсказками	самостоятельно объясняет и отвечает на вопросы

Итоговый балл: ____ / 10. Решение: ☐ зачёт ☐ требуется доработка. Комментарий педагога: _____.