

**Частное образовательное учреждение дополнительного
профессионального образования «Учебный центр»**

Утверждено:

Решением педагогического совета
протокол № 1 от «13» 09 2021



/Сипкин П.Н./

Дополнительное образование для детей

Курс программирования Python Pro (1 год обучения)

Направленность: техническая

Уровень программы: ознакомительный

Возраст учащихся: 14-17 лет

Срок реализации: 1 учебный год (64 академических часа)

Составитель:

Мария Липчанская

Ирина Курзенко

Рассмотрено:

Директор – Сипкин Павел
Николаевич

Методист – Наговская Нелли
Шамильевна

Абакан, 2021

Пояснительная записка

1.1. Направленность и уровень Программы

Информационные технологии — одна из наиболее быстро развивающихся сфер экономики. Ежегодно появляется множество новых вакансий в области разработки десктопных и мобильных приложений, создания сайтов, исследований в сфере искусственного интеллекта и анализа данных, и многих других. Статистика показывает, что востребованность специалистов в перечисленных сферах будет расти в последующие годы.

Python является одним из популярных и быстро развивающихся языков программирования, который используется для решения всех перечисленных выше задач. Кроме того, по мнению экспертов, Python обладает простым синтаксисом и прост в изучении. Именно поэтому данный язык рекомендуется как один из стартовых в освоении программирования.

Еще одной важной задачей в подготовке IT-специалистов является развитие таких Soft Skills, как умение работать в команде, планирование, креативность, критическое мышление, навыки проведения презентаций и публичных выступлений.

Курс программирования «Python Pro» позволяет учащимся познакомиться с профессиональным языком программирования Python и получить опыт решения задач, относящихся к разным рынкам и сферам IT, начать формировать портфолио разработчика и развивать перечисленные выше soft skills.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у учащихся среднего и старшего школьного возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами с использованием разных типов платформ и интерфейсов. А также в развитии у учащихся умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формированию навыков сотрудничества и коллективной работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования логического, алгоритмического, критического мышления и навыков проектной деятельности.

Программа составлена с учетом задач национального проекта «Цифровая экономика» (федерального проекта «Кадры для цифровой экономики») и национального проекта «Образование» (федерального проекта «Молодые профессионалы»): создание условий для формирования у учащихся основ цифровых компетенций необходимых для дальнейшего участия в разработке и поддержке инфраструктур передачи, обработки и хранения данных, а также

цифровых продуктов для граждан, бизнеса и власти; разработке, внедрении цифровых технологий и платформенных решений.

1.2. Актуальность

В рамках курса учащиеся осваивают базовые принципы программирования на языке Python. За время курса ребята создадут несколько собственных проектов и попробуют себя в роли наиболее востребованных IT-профессионалов — от разработчиков игр до специалистов по машинному обучению (к концу второго года обучения).

Особенность курса проявляется в том, что он направлен на проектную деятельность: проектные модули занимают более 60% курса. Изучение теоретического материала сконцентрировано в первых 3 модулях курса. Результатом каждого такого модуля базового (теоретического) блока является программа, самостоятельно написанная учеником, в которой используются все инструменты, изученные в модуле. В проектной блоке в каждом модуле учащиеся знакомятся с функционалом одной из профессиональных библиотек, предназначенных для решения определенного круга задач. В программу обучения включены библиотека для разработки игр PyGame и библиотека для создания пользовательского интерфейса PyQt5. Начиная с библиотеки PyQt5 при выборе тематики итогового проекта учащимся будет предлагаться работа в рамках одного из трех треков: естественнонаучного, гуманитарного, бизнес-экономического. Работа над проектами в рамках треков будет строиться с элементами геймификации и включать следующие этапы:

- *Тендер* — учащимся предлагаются технические задания от трех заказчиков. На этом этапе нужно выбрать проект, над реализацией которого будет работать учащийся или команда.
- *Планирование* — разработка концепции и плана работ.
- *Прототипирование* — концепция презентуется классу и обсуждается с другими командами и преподавателем.
- *Разработка (часть 1)* — учащийся работает над проектом индивидуально или в команде.
- *Тестирование* — учащиеся тестируют проекты друг друга.
- *Разработка (часть 2)* — доработка проекта на основе полученной обратной связи.
- *Презентация* — презентация проекта классу и (по желанию и обстоятельствам) родителям, местной прессе или блогерам, руководству школы, на базе которой проводятся занятия и т. п.

Использование компьютерных технологий в работе с детьми среднего и старшего школьного возраста является стремительно развивающейся методикой в образовании во всем мире. С ее помощью можно более эффективно решать образовательные задачи, которые будут способствовать качественному улучшению обучения ребенка в школе.

Актуальность программы заключается в:

- формировании основных навыков программирования на языке Python;
- формировании навыков работы с базами данных, использования библиотеки PyGame и PyQt;
- формировании и развитии навыков работы с информацией в условиях модернизации образования;
- формировании навыков работы с графикой и разработки пользовательских интерфейсов;
- формировании навыков предоставления и получения обратной связи;
- формировании навыков управления проектами;
- развитии цифровой и функциональной грамотности;
- развитии алгоритмического и креативного мышления, творчества через создание собственных проектов с использованием цифровых ресурсов;
- расширении кругозора школьника.

Новизна программы заключается в технической направленности обучения, которое проявляется в знакомстве с современными направлениями программирования, что способствует развитию информационной культуры. В отличие от других курсов программирования Python Pro предоставляет учащимся возможность с одной стороны освоить все базовые понятия программирования, используя их непосредственно в разработке приложений; с другой стороны — познакомиться с современным представлением о работе программиста, изучая разнообразные современные технологии и библиотеки для их применения; с третьей стороны — выбирать наиболее близкие ученикам области для реализации полученных знаний. Проекты, выполняемые на уроках, охватывают разные области интересов, в том числе школьные предметы. Также ученики развивают умение самопрезентации и презентации своего проекта или идеи как в группе, так и перед учителями и родителями.

Авторское воплощение замысла курса заключается в том, что ученики используют инструменты профессиональной разработки для реализации собственных проектов и формирования портфолио разработчика. Для этого сначала ученик получает базовые знания о синтаксисе языка и алгоритмах, затем учится проектной деятельности, после чего реализует проекты в области своих увлечений. При этом базовая часть также подразумевает возможность выразить свои интересы. В базовой части проекты выполняются преимущественно индивидуально, чтобы учащиеся сначала научились работать над ними, затем уже — работать в группе.

Эволюция компьютеров и программного обеспечения привела к относительной простоте освоения некоторых профессиональных инструментов, благодаря чему программирование может изучаться на серьезном уровне уже в школьном возрасте.

1.3. Цель и задачи Программы

Цель курса — способствовать формированию у учащихся понимания, как использовать язык программирования Python в качестве инструмента для

реализации идей в области своих личных или будущих профессиональных интересов.

Достижение поставленной цели складывается из выполнения следующих задач:

Обучающие:

1. Познакомить с базовыми понятиями основ программирования.
2. Познакомить с особенностями синтаксиса языка программирования Python.
3. Познакомить с основами парадигм структурного и объектно-ориентированного программирования.
4. Сформировать навык правильного оформления кода.
5. Сформировать навык использования итерационного подхода при решении различных задач.
6. Формировать навык работы со структурами данных с помощью языка программирования Python.
7. Формировать навык изучения библиотек, получить представление о работе с библиотеками PyGame и PyQt5.
8. Формировать навык решения задач и разработки проектов с помощью Python.
9. Формировать навык разработки игр с помощью языка программирования Python.
10. Формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

1. Формировать и развивать логическое, алгоритмическое и критическое мышление.
2. Развивать навык публичного выступления и презентации.
3. Развитие креативных способностей и умений учиться.
4. Развивать навык планирования деятельности и основ тайм-менеджмента.
5. Развивать функциональную грамотность.
6. Развивать мотивацию успеха, готовность к действиям в новых условиях и нестандартных ситуациях.
7. Развивать коммуникативные способности и умение работать в команде.
8. Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.
9. Совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл используемых терминов, уметь задавать вопросы, отвечать на них.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект.

2. Формировать умение давать качественную обратную связь и реагировать на нее.

3. Формировать информационную культуру: умение целенаправленно использовать современные технические средства и процессы для получения, обработки и передачи информации.

4. Повышение компетентности учащихся в области планирования карьеры.

Данные задачи не только способствуют формированию предметных и метапредметных компетенций, но и формируют мировоззрение школьника, раскрывают значение soft skills, учат грамотному использованию компьютера и готовят к жизни в цифровом обществе.

1.4. Формы и режим занятий

Занятия проходят 1 раз в неделю и длятся 2 академических часа. Дети занимаются в классе с учителем, самостоятельно работая за компьютером (непрерывно — не более 30 минут), занятия разделены перерывом. Основные формы работы — индивидуальная, фронтальная, в т.ч. через организацию групповых дискуссий.

Формы обучения:

- Обучение от общего к частному.
- Поощрение вопросов и свободных высказываний по теме.
- Уважение и внимание к каждому ученику.
- Создание мотивационной среды обучения.
- Создание условий для дискуссий и развития мышления учеников при достижении учебных целей вместо простого одностороннего объяснения темы преподавателем.

Учебное занятие состоит из следующих блоков:

1. Повторение и актуализация знаний (5 минут).
2. Изучение нового материала (15 минут).
3. Практика на платформе (15 минут).
4. Рефлексия: подведение промежуточных итогов (5 минут).
5. Разминка (10 минут).
6. Изучение нового материала (15 минут).
7. Практика на платформе (20 минут).
8. Рефлексия: подведение итогов занятия (5 минут).

Занятие проектного типа состоит из следующих блоков:

1. Повторение (5 минут).
2. Планирование проектов (5 минут).
3. Программирование (25 минут).

4. Подведение промежуточных итогов (5 минут).
5. Разминка (10 минут).
6. Программирование (25 минут).
7. Научпоп (10 минут).
8. Рефлексия (5 минут).

Ученики могут давать обратную связь по проектам друг друга, для чего используются этапы повторения, подведения итогов и рефлексии, также предусмотрены занятия с возможностью презентовать собственную работу.

1.5. Срок реализации Программы

Содержание курса программирования Python Pro для обучающихся 14–17 лет рассчитано на обучение в течение 8 месяцев (32 занятия — 64 академических часа).

1.6. Планируемые результаты

Достижение цели и задач образовательной программы предполагает получение следующих результатов:

Планируемые результаты	Способ достижения	Критерий достижения образовательного результата
Предметные навыки		
Знакомство с синтаксисом языка программирования Python.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно пишет программы, используя язык программирования Python.
Развитие умений применять Python для решения задач практического характера и задач из смежных дисциплин.	Выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно пишет программы на языке Python, решающие задачи практического характера.
Развитие умения выбрать подходящую задаче структуру данных.	В начале проекта при планировании ученики в том числе рассказывают, какие структуры данных собираются применить и доказывают, почему это лучшее решение.	Ученик для выполнения проекта без подсказки учителя выбирает оптимальную из известных ему структуру данных.
Следование стандарту при оформлении кода.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе.	Ученик оформляет код в соответствии с общепринятыми стандартами и использует комментарии.
Умение тестировать программу и находить в ней ошибки.	Выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно находит ошибки в своем и чужом коде.
Умение использовать чужой код при разработке: анализировать и изменять.	Выполнение заданий на платформе.	Ученик самостоятельно вносит изменения в готовый код и использует фрагмент готового кода в процессе

		реализации собственной программы.
Использование итерационного подхода.	Выполнение заданий на платформе.	Ученик пишет программы, разделяя задачу на отдельные части.

Личностные навыки		
Освоение социальной роли обучающегося и формирование личностного смысла учения.	Демонстрация связи между способностью выполнить интересную задачу и наличием/отсутствием соответствующих знаний.	За отведенное время ученик пытается не только выполнить базовые уровни, но и приступить к бонусным.
Развитие навыков сотрудничества со взрослыми и сверстниками и умения находить выходы из спорных ситуаций.	Использование программирования как способа показать преимущества работы в команде.	Ученик не боится просить помощь и сам пытается помогать одноклассникам и учителю.
Овладение начальными навыками адаптации в динамично изменяющемся и развивающемся мире.	Обсуждение научно-популярных материалов, демонстрирующих необходимость не просто овладения предметными навыками, но и развития умения учиться.	При работе над проектом ученик сначала пытается найти информацию в Интернете или теоретической справке на платформе и только при неудаче задает вопрос учителю.
Умение аргументировать свою позицию.	Выступление во время защиты проектов, ответы на вопросы.	Ученик обосновывает подход к решению проблемы, используя цепочки аргументов. Помнит, как использовались логические связи между ними. Отвечает на критику аргументами, а не эмоциями.
Умение решать задачи, с которыми раньше не сталкивался.	Ситуации успеха при решении незнакомых задач: вместо того, чтобы дать пример, затем повторить, сначала даем задачу и через подсказки помогаем дойти до решения.	Ученик, сталкиваясь с новой задачей, ищет решение: задает вопросы, проясняющие структуру задачи; находит знакомые подзадачи и аналогии; разделяет составные части задачи. При возникновении проблемы предлагает способы ее решения или понимает, что текущих знаний для решения проблемы не хватает и предлагает способы эти знания получить.
Понимание того, что ошибки — это инструмент обучения.	Часть заданий подразумевают, что надо попробовать несколько методов, один из которых заработает, единственно верный вариант не дается заранее.	Ученик пробует свои идеи, не ожидая разрешения. При обнаружении ошибок старается исправлять их самостоятельно.

Понимание того, что компьютер — уникальный инструмент, с помощью которого можно добиться многого.	Акцентирование внимания на том, что с помощью компьютера делаются вещи, которые без него делались бы месяцами (или были бы невозможны вообще).	Ученик в соответствующих тестовых заданиях ставит компьютеру самую высокую оценку полезности.
---	--	---

Метапредметные навыки		
Развитие и формирование учебных действий.	Создание благоприятных условий для участия в диалоге, в коллективном обсуждении. Строится продуктивное взаимопонимание со сверстниками и взрослыми в процессе коллективной деятельности.	Ученик легко общается, не боится просить помощь или оказать ее другим. Ученик способен кооперироваться с другими, чтобы достичь цели.
Умение презентовать свою работу.	Презентация индивидуальных проектов. Учитель и другие ученики дают обратную связь. Учитель также дает советы, каким образом это лучше делать.	Ученик во время презентации своих проектов пользуется вниманием аудитории.
Развитие критического и алгоритмического мышления учеников.	Выполнение логических операций: сравнение, анализ, синтез, обобщение, установление аналогий на уроке. Анализ любой получаемой информации, составление плана действий для решения задачи.	Ученик строит логическую цепь рассуждений. Управляет своей деятельностью.
Развитие творческих способностей учеников.	Планирование и создание проекта по сценарию. Использование инструментов цифрового сервиса для воплощения своего проекта.	Ученик придумывает, обсуждает, планирует и реализует свой проект.
Умение управлять своим временем: устанавливать приоритеты, умеет отказываться от неважных задач.	Выполнение заданий, направленных на получения опыта ведения проектов.	Ученик умеет планировать свое время, определять приоритет задач, благодаря чему успевает сделать проект в срок и доволен своим результатом.
Умение совместно с другими распределять задания в группе.	Выполнение групповых проектов.	Ученик без помощи преподавателя договаривается с другими участниками команды, кто что делает.
Умеет давать и получать обратную связь.	Презентация проектов.	Ученик в ходе комментирования работы другого избегает оценочных суждений, выделяет как сильные, так и слабые стороны.

Умеет планировать проекты.	Выполнение проектов: в начале выполнения проектов зарисовывается его план, учитель показывает, как это лучше делать.	Ученик раскладывает проект на составные элементы, затем выстраивает их в последовательность шагов.
----------------------------	--	--

Содержание Программы

2.1. Учебный (тематический) план

№ п/п	Наименование раздела/темы	Количество часов			Форма аттестации (контроля)
		всего	теория	практика	
Модуль 1. Введение в Python		10			
1 урок	Знакомство с языком	2	1	1	Задания на платформе
2 урок	Типы данных	4	2	2	Задания на платформе
3 урок	Условный оператор	4	2	2	Задания на платформе
Модуль 2. Циклы и списки		14			
4 урок	Цикл while	2	1	1	Задания на платформе
5 урок	Символы и строки	2	1	1	Задания на платформе
6 урок	Списки	2	1	1	Задания на платформе
7 урок	Цикл for	2	1	1	Задания на платформе
8 урок	Сортировки	2	1	1	Задания на платформе
9 урок	Словари и множества	2	1	1	Задания на платформе
10 урок	Проект «Шифр»	2	1	1	Защита проекта
Модуль 3. Функции и классы		10			
11 урок	Создание функций	2	1	1	Задания на платформе
12 урок	Создание модулей. Модули math, random	2	1	1	Задания на платформе
13 урок	Классы	2	1	1	Задания на платформе
14 урок	Проект «База данных»	4	2	2	Защита проекта
Модуль 4. 2D игры		22			
15 урок	Знакомство с библиотекой PyGame	2	1	1	Задания на платформе

16 урок	Учебный проект «Лабиринт»	6	3	3	Защита проекта
17 урок	Индивидуальный проект «Шутер»	8	4	4	Защита проекта
18 урок	Групповой проект «Аркада»	6	3	3	Защита проекта
Модуль 5. Десктопные приложения		8			
19 урок	Знакомство с PyQt	2	1	1	Задания на платформе
20 урок	Групповой проект по выбору	4	2	2	Защита проекта
21 урок	Выпускной	2	1	1	Защита проекта

2.2. Содержание учебного (тематического) плана

Модуль 1. Введение в Python

1.1. Знакомство с языком

- Язык программирования Python. Знакомство с синтаксисом языка. Переменная. Значение переменной. Оператор присваивания. Функция. Вызов функции. Параметры. Функция ввода `input()`. Функция вывода `print()`. Необходимость ввода и вывода данных.

- Разработка программы, осуществляющей ввод-вывод данных, ввод команд с клавиатуры.

1.2. Типы данных

- Данные. Типы данных. Переменная. Числовые типы данных. Строковый тип данных. Приведение к новому типу данных. Приоритет операций. Выражение.

- Разработка программы, осуществляющей обработку переменных числового типа с использованием арифметических операций, в т. ч. и целочисленных.

1.3. Условный оператор

- Логический тип данных. Утверждение. Условие. Условный оператор. Синтаксис условного оператора. Логические операторы. Вложенный условный оператор. Уровни вложенности.

- Разработка текстового квеста, содержащего не менее 3 вопросов, т. о. чтобы сюжет развивался в зависимости от принятого игроком решения.

Модуль 2. Циклы и списки

2.1. Цикл while

- Цикл. Цикл `while`. Переменная-счетчик. Цикл с накоплением результата вычислений в переменной.

- Разработка программы, осуществляющей обработку данных с помощью цикла `while`.

2.2. Символы и строки

- Строки. Символы. Индекс символа. Кодировка символов. Методы для работы со строками `len()`, `find()` и `rfind()`, `replace()`, `count()`. Извлечение из строки отдельных символов и набора символов.

- Разработка программы, осуществляющей обработку строк.

2.3. Списки

- Список. Элементы списка. Индекс элемента. Создание пустого списка. Заполнение списка. Методы для работы со списками `len()`, `append()`, `split()`, `join()`, `list(map(int, input().split()))`. Извлечение отдельных элементов списка.

- Разработка программы, осуществляющей обработку списка.

2.4. Цикл for

- Цикл `for`. Функция `range()`. Отличие цикла `for` от цикла `while`. Цикл с вложенными условными операторами.

- Разработка программы, осуществляющей обработку данных с помощью цикла `for`.

2.5. Сортировки

- Сортировка. Алгоритмы сортировки. Вложенные циклы. Расчет сложности алгоритмов. Графы.

- Разработка программы, осуществляющей сортировку.

2.6. Словари и множества

- Словарь. Ключ. Множество. Принципы хранения элементов в множествах и словарях. Создание и заполнение множества и словаря.

- Разработка программ, содержащих словарь и множества. Получение доступа и использование элементов словаря с помощью ключа.

2.7. Проект «Шифр»

- Реализация проекта «База данных»: создание программы, которая умеет шифровать и дешифровать сообщение, используя придуманный шифр.

Модуль 3. Функции и классы

3.1. Создание функций

- Функция. Оператор `def`. Параметры функции. Значение функции. Вызов функции. Синтаксис создания функций. Локальные и глобальные переменные.

- Разработка программ, осуществляющих обработку данных с помощью функций, создание функций.

3.2. Создание модулей. Модули `math`, `random`

- Понятие модуля. Встроенные модули стандартной библиотеки. Создание собственного модуля. Подключение модуля. Модуль `random`. Модуль `math`.

- Разработка программ, осуществляющих обработку данных с помощью модулей.

3.3. Классы

- *Объекты и классы. Конструктор класса. Свойства и методы объектов. Создание объектов.*

- *Разработка программ, осуществляющих обработку классов.*

3.4. Проект «База данных»

- *Реализация проекта «База данных»: создание базы данных (учеников, продуктов, игр и пр. на выбор), сформированной на основе классов.*

Модуль 4. 2D игры

4.1. Знакомство с библиотекой PyGame

- *Библиотека PyGame. Наследование. . Класс-наследник. Объекты. Использование готовых классов в своих программах. Создание объектов. Использование событий.*

- *Разработка программы с помощью библиотеки PyGame.*

4.2. Учебный проект «Лабиринт»

- *Реализация учебного проекта «Лабиринт».*

4.3. Индивидуальный проект «Шутер»

- *Реализация индивидуального проекта «Шутер».*

4.4. Групповой проект «Аркада»

- *Реализация группового проекта «Аркада».*

Модуль 5. Desktopные приложения

5.1. Знакомство с PyQt

- *Пользовательский интерфейс. Знакомство с PyQt. Объекты “приложение”, “виджет”. Знакомство с различными видами виджетов. Создание своих виджетов.*

- *Разработка пользовательского интерфейса с помощью PyQt.*

5.2. Групповой проект по выбору

- *Выполнение группового проекта по выбору: «Тест здоровья», «Финансовый калькулятор», «Психологический тест».*

5.3. Выпускной

- *Обобщение и систематизация пройденного материала в игровой форме.*

2.3 Формы аттестации и оценочные материалы

Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях:

- **на каждом занятии:** опрос, выполнение заданий на платформе, реализация проектов, самоконтроль ученика либо взаимоконтроль учеников в парах или мини-группах, общее подведение итогов занятия;
- **в конце каждого модуля:** проведение презентации (по желанию) финальных проектов модуля и их оценка.

Для контроля сформированности результатов освоения программы на платформе используются задания на программирование (как написание программ с нуля, так и исправление ошибок в предложенной программе), клавиатурный тренажер и следующие механики тестирования:

- классический тест (выбор одного или нескольких правильных ответов),
- заполнение пропусков,
- классификация,
- сопоставление,
- сортировка,
- ответ в свободной форме, в т. ч. прикрепление файла.

Задания на платформе представлены как с автопроверкой (все задания на тему базовых понятий проверяются автоматически системой), так и с возможностью свободного решения; в проектных заданиях имеются чек-листы для преподавателей и учеников. Преподаватель имеет доступ ко всем кодам учеников (как записанных на онлайн-платформе, так и в системе Visual Studio Code), помогает находить ошибки и проводить ревью кода (выборочная оценка кода учеников использование оптимальных конструкций и структур данных, и на соблюдение стандартов его написания). В учебном плане предусмотрены конференции и круглые столы, на которых возможна взаимопроверка, собирается обратная связь учеников, может проводиться анкетирование родителей.

3. Организационно-педагогические условия реализации Программы

3.1. Организационно-педагогические условия реализации Программы содержат:

материально-технические условия реализации Программы:

- учебный кабинет с местами для 12 учеников;
- компьютеры для учеников (1 ученик — 1 компьютер) и компьютер для учителя (требования к компьютерам: ОС Windows версии 7, 8 или 10, либо Mac OS версии не ниже 10.10, либо Linux — Ubuntu Desktop не ниже 14.04 или Debian 7; оперативная память объемом от 4 Гб, рекомендуемый объем от 8Гб; процессор Intel серии Core I3, 5, 7 не младше третьего поколения, или серий Pentium Silver/Gold, либо AMD серии Ryzen или старшие модели серий A10 и A12);
- стационарный доступ в Интернет с компьютеров учеников и учителя (рекомендуется от 30 Мбит/с);
- веб-браузер на основе Chromium (рекомендуется Google Chrome), обновленный до последней версии;
- проектор, подключенный к компьютеру для учителя, выводящий изображение на экран;
- распечатанные логины и пароли учеников для доступа на платформу

- (учитель имеет доступ к общему списку логинов и паролей своей группы в электронном виде);
- установленная на компьютерах среда разработки Visual Studio Code с дополнением от Алгоритмики, которое позволяет организовывать учебный процесс работы над проектами (инструкция по установке дается учителю вместе с другими учебными материалами);
 - канцелярия: бумага А4, ручки/карандаши;
 - доска или флипчарт, мел или маркер.

учебно-методическое и информационное обеспечение Программы:

- поурочные методические рекомендации к занятиям;
- тематические презентации;
- задания на платформе с автопроверкой/со свободным решением;
- бонусные задания на платформе;
- задачи со свободным решением для реализации в среде программирования Visual Studio Code, интегрированной в платформу.

Приложение № 1 к Программе

Календарный учебный график

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Количество часов	Тема занятия	Место проведения	Форма контроля
1				Интерактивное занятие	2	Знакомство с языком		Задания на платформе
2				Интерактивное занятие	2	Типы данных		Задания на платформе
3				Интерактивное занятие	2	Типы данных		Задания на платформе
4				Интерактивное занятие	2	Условный оператор		Задания на платформе
5				Интерактивное занятие	2	Условный оператор		Задания на платформе
6				Интерактивное занятие	2	Цикл while		Задания на платформе
7				Интерактивное занятие	2	Символы и строки		Задания на платформе
8				Интерактивное занятие	2	Списки		Задания на платформе
9				Интерактивное занятие	2	Цикл for		Задания на платформе
10				Интерактивное занятие	2	Сортировки		Задания на платформе
11				Интерактивное занятие	2	Словари и множества		Задания на платформе
12				Интерактивное занятие	2	Проект «Шифр»		Защита проекта
13				Интерактивное занятие	2	Создание функций		Задания на платформе
14				Интерактивное занятие	2	Создание модулей. Модули math, random		Задания на платформе
15				Интерактивное занятие	2	Классы		Задания на платформе
16				Интерактивное занятие	2	Проект «База данных»		Задания на платформе
17				Интерактивное занятие	2	Проект «База данных»		Защита проекта
18				Интерактивное занятие	2	Знакомство с библиотекой PyGame		Задания на платформе
19				Интерактивное занятие	2	Учебный проект «Лабиринт»		Задания на платформе
20				Интерактивное занятие	2	Учебный проект «Лабиринт»		Задания на платформе

21				Интерактивное занятие	2	Учебный проект «Лабиринт»		Защита проекта
22				Интерактивное занятие	2	Индивидуальный проект «Шутер»		Задания на платформе
23				Интерактивное занятие	2	Индивидуальный проект «Шутер»		Задания на платформе
24				Интерактивное занятие	2	Индивидуальный проект «Шутер»		Задания на платформе
25				Интерактивное занятие	2	Индивидуальный проект «Шутер»		Защита проекта
26				Интерактивное занятие	2	Групповой проект «Аркада»		Задания на платформе
27				Интерактивное занятие	2	Групповой проект «Аркада»		Задания на платформе
28				Интерактивное занятие	2	Групповой проект «Аркада»		Защита проекта
29				Интерактивное занятие	2	Знакомство с PyQT		Задания на платформе
30				Интерактивное занятие	2	Групповой проект по выбору		Задания на платформе
31				Интерактивное занятие	2	Групповой проект по выбору		Защита проекта
32				Интерактивное занятие	2	Выпускной		Защита проекта

Список литературы

1. Кларин М.В. Инновационные модели обучения: Исследование мирового опыта. Монография. 2-е издание. — М.: ЛУЧ, 2018. — 640 с.
2. Лапчик М.П., Семакин И.Г., Хеннер Е.К., Самылкина Н.Н. Теория и методика обучения информатике. — М.: Академия, 2008. — 592 с.
3. Федеральный государственный образовательный стандарт основного общего образования (5-9 кл.) [Электронный ресурс] Режим доступа: минобрнауки.рф/документы/543.
4. Федеральный закон «Об образовании в РФ» [Электронный ресурс] Режим доступа: consultant.ru/document/cons_doc_LAW_140174/
5. Актуальные проблемы методики обучения информатике в современной школе : материалы Международной научно-практической интернет-конференции, г. Москва, 24–26 апреля 2018 г. / под ред. Л. Л. Босовой, Н. К. Нателаури ; Московский педагогический государственный университет. Кафедра теории и методики обучения информатике [Электронное издание]. — Москва: МПГУ, 2018. — 222 с.
6. А.В. Хуторской. Компетентностный подход в обучении. Научно-методическое пособие. — М.: Издательство «Эйдос»; Издательство Института образования человека, 2013. — 73 с.
7. А.А. Вербицкий, О.Г. Ларионова: Личностный и компетентностный подходы в образовании: проблемы интеграции. Монография. — М.: Издательство «Логос», 2017. — 336 с.
8. К. Вольдерман, К. Стили, К. Квигли, М. Гудфеллоу, Д. Маккафферти, Дж. Вудкок. Программирование на Python: иллюстрированное руководство для детей. — М.: Манн, Иванов и Фербер, 2018. — 224 с.: илл.
9. Дж. Бриггс. Python для детей. Самоучитель по программированию. — М.: Манн, Иванов и Фербер, 2020.