

Индивидуальный предприниматель Кучубаев Тимур Равилевич



Утверждено
Кучубаев Т.Р.

Дополнительная общеобразовательная программа –

Дополнительная общеразвивающая программа

«Курс программирования Python Старт»

2 год обучения

Направленность: техническая

Уровень программы: ознакомительный

Возраст учащихся: 12–14 лет

Срок реализации: 1 учебный год (64 академических часа)

Авторы-составители:

Кучубаев Тимур Равилевич

г. Пермь, 2023 г.

1. Пояснительная записка

1.1. Направленность и уровень Программы

Программирование давно стало частью повседневной жизни человека. Каждый из нас так или иначе имеет к нему отношение, так как информационные технологии давно проникли во все сферы экономики и повседневной рутины от электронных очередей и онлайн-шопинга до вечернего чтения и прослушивания музыки, не говоря уже о ежедневном общении с людьми. При этом потребность в специалистах, которые способны обеспечить работу информационных систем, которые нас окружают только растет.

Python - один из наиболее доступных и понятных языков программирования - используется при разработке настольных и мобильных приложений, при анализе данных, при создании сайтов и т.д. Данный язык является одним из наиболее популярных среди разработчиков и потому очень актуален для начала погружения в эту профессиональную область.

Кроме того, при изучении и регулярной практике в программировании неизбежно развиваются такие навыки как способность самостоятельно учиться и осуществлять контроль своей деятельности, внимательность, критическое и алгоритмическое мышление. А также умение работать в команде, креативность и навыки презентации результатов своей работы.

Курс программирования «Python Старт» (2 год обучения) позволяет учащимся познакомиться с профессиональным языком программирования Python через захватывающее путешествие в разработку игр и приложений. Современный мир предъявляет новые требования к молодому поколению, вступающему в жизнь, так как будущее сегодняшних детей — это цифровое общество.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у учащихся среднего школьного возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также

развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективной работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного 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; разработка, внедрение цифровых технологий и платформенных решений.

1.2. Актуальность

Обучение осуществляется в рамках ролевой игры, сюжетная линия которой охватывает весь курс. Курс построен таким образом, что дает возможность учащимся реализовать несколько больших проектов в рамках работы на позиции разработчика в большой IT-компании. Данные проекты не только способствуют формированию предметных компетенций, но еще и формируют взгляд ученика на мир, способствуют развитию широкого спектра метапредметных и личностных навыков, в частности грамотному использованию компьютера и готовят к жизни в цифровом обществе.

В рамках курса учащиеся вспомнят базовые принципы программирования на языке Python, а также погрузятся в работу с профессиональной кроссплатформенной средой программирования Visual Studio Code, в которой создадут несколько оконных приложений и игр.

Особенность курса проявляется в том, что весь изученный теоретический материал сразу практикуется в рамках реализации учебного проекта основанного на адаптированных задачах из реальной жизни. Кроме того, навыки применяются не по отдельности, а сразу в совокупности с изученным ранее материалом, что особенно эффективно в работе над

проектами. Результат работы, воплощающий все полученные учащимися знания и навыки, ребята защищают перед другими учениками, педагогами и родителями.

Использование компьютерных технологий в работе с детьми среднего 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 навыков работы с различными библиотеками языка Python;
- формировании навыков использования библиотек PyGame и PyQt;
- формировании навыков планирования и разработки пользовательского интерфейса;
- формировании навыков продумывания и реализации последовательности игровых событий;
- формирование навыка работы с в профессиональном сообществе разработчиков;
- формировании навыков предоставления и получения обратной связи;
- необходимости развития мышления и творческого потенциала школьников через создание собственных проектов;
- формировании навыков управления проектами;
- развитии цифровой и функциональной грамотности.

Новизна программы заключается в превалировании прикладного аспекта над фундаментальным, что позволяет изучать программирование с нуля без значительной опоры на математику, демонстрировать применение полученных знаний в решении задач из реальной жизни и мотивировать учеников с разным уровнем подготовки. В отличие от других курсов программирования Python Start предоставляет учащимся возможность думать и действовать как программисты, а не просто решать стандартный набор задач. Более того, проекты, выполняемые на уроках, охватывают разные аспекты предметных областей, в том числе, выходящие за рамки школьной программы. Также ученики развивают умение самопрезентации и презентации своих проектов или идей как в группе, так и перед учителями и родителями.

Авторское воплощение замысла курса заключается в том, что ученики используют реальные инструменты разработки для реализации собственных проектов и формирования своего портфолио разработчика. Для этого сначала ученик комплексно и интенсивно повторяет основы программирования (алгоритмы, структурное программирование, использование библиотек в процедурном стиле), а затем погружается в работу с профессиональными библиотеками, которые расширяют спектр его возможностей как программиста за пределы стандартного и базового функционала языка. Проекты, которые ученики создают выполняются преимущественно индивидуально, благодаря чему каждый ученик может отразить в работе свои интересы. В том числе это позволяет сначала отточить навык работы над проектом в принципе, а затем — работу над ним в команде. Эволюция программного обеспечения и языков программирования привела к достаточной простоте их освоения для самых неподготовленных пользователей, в том числе школьников.

1.3. Цель и задачи Программы

Цель курса — способствовать формированию алгоритмического мышления и навыков проектной деятельности, а также реализации идей в области своих интересов через использование языка программирования Python.

Достижение поставленной цели складывается из выполнения следующих задач:

Обучающие:

1. Сформировать навык правильного оформления кода.
2. Формировать навык решения задач и разработки проектов с помощью Python.
3. Познакомить с профессиональными библиотеками языка Python (PyQt, PyGame, Pillow) и применить их на практике.
4. Сформировать навык использования итеративного подхода при решении различных задач.
5. Сформировать навык работы в профессиональном сообществе (на примере сообщества учеников Алгоритмики "Зал славы" и международного сервиса для публикации IT-проектов GitHub).

6. Познакомить с профессиональными инструментами разработчика (работа с текстовыми и графическими файлами, использование игровой библиотеки, GitHub).

7. Формировать навык разработки игр с помощью языка программирования Python.

8. Формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

1. Формировать и развивать логическое, алгоритмическое, критическое и креативное мышление.

2. Формировать навык публичного выступления и презентации.

3. Формировать навык планирования деятельности и основ тайм-менеджмента.

4. Формировать и расширять словарный запас на английском языке.

5. Формировать функциональную грамотность.

6. Развивать мышление, память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.

7. Совершенствовать речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, 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.

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 20 минут), занятие разделено перерывом. Основные формы работы — индивидуальная, фронтальная.

Специфика форм обучения:

- Обучение от общего к частному (дедуктивный метод) и от частного к общему (индуктивный).
- 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.

1.5. Срок реализации Программы

Курс программирования «Python Start» (2 год обучения) для обучающихся 12–14 лет рассчитан на обучение в течение 1 года (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		
Формирование базовых навыков работы с основными инструментами разработчика на Python.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе с постепенным увеличением сложности.	Ученик оперирует базовыми профессиональными инструментами разработчика при работе над проектами.
Знакомство с библиотеками языка программирования Python: PyQt, Pygame, Pillow.	Изучение теоретического материала совместно с учителем, выполнение заданий в профессиональной среде с расширением образовательной	Ученик самостоятельно пишет программы, используя библиотеки языка программирования Python: PyQt, Pygame, Pillow.

	платформы Алгоритмики с постепенным увеличением сложности.	
Развитие умений применять Python для решения задач практического характера и задач из смежных дисциплин.	Выполнение заданий в профессиональной среде с расширением образовательной платформы Алгоритмики с постепенным увеличением сложности.	Ученик самостоятельно пишет программы на языке Python, решающие задачи практического характера.
Знакомство с GitHub и его использование для публикации проектов.	Изучение технологии работы на GitHub и ее применение на практике.	Ученик создает репозиторий со своим итоговым проектом и делится ссылкой на него.
Следование стандарту при оформлении кода.	Изучение теоретического материала совместно с учителем, выполнение заданий в профессиональной среде с расширением образовательной платформы Алгоритмики.	Ученик оформляет код в соответствии с общепринятыми стандартами и использует комментарии.
Умение тестировать программу и находить в ней ошибки.	Выполнение проектных заданий с постепенным увеличением сложности.	Ученик самостоятельно определяет и исправляет ошибки в своем и чужом коде.
Использование итерационного подхода.	Выполнение заданий в профессиональной среде с расширением	Ученик пишет программы, разделяя код на отдельные части.

	образовательной платформы Алгоритмики.	
--	---	--

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 выполнить не только обязательную, но и дополнительную часть проектного задания.
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 теоретической справке на платформе и только при неудаче задает вопрос учителю.

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, 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.

2. Содержание Программы

2.1. Учебный (тематический) план

№ п/п	Наименование раздела/темы	Количество часов			Форма аттестации (контроля)
		всего	теория	практика	
Модуль 1. Структуры данных		8			
1	Повторение. Обработка исключений	2	1	1	Задания на платформе
2.	Повторение. Списки	2	1	1	Задания на платформе
3.	Словари	2	1	1	Задания на платформе
4.	Вложенные структуры данных	2	1	1	Задания на платформе
Модуль 2. Разработка оконных приложений		12			
5.	Классы. Введение в PyQt	2	1	1	Задания в среде программирования с расширением учебной платформы Алгоритмики
6.	Проектирование интерфейса	2	1	1	
7.	Последовательное создание оконного приложения Memory Card	8	4	4	
Модуль 3. Работа с файлами		8			
8.	Основы работы с файлами	2	1	1	Задания в среде программирования с расширением учебной платформы Алгоритмики
9.	Создание оконного приложения «Умные заметки» с поиском записей по тегам	6	3	3	
Модуль 4. Автоматическая обработка изображений		8			
10.	Основы обработки изображений	2	1	1	Задания в среде программирования с расширением учебной платформы Алгоритмики
11.	Создание приложения Easy Editor для обработки фотографий	6	3	3	
Модуль 5. Продвинутая разработка игр на PyGame		18			
12.	Основы создания игр	2	1	1	Задания в среде программирования с расширением учебной платформы Алгоритмики
13.	Создание игры Лабиринт	6	3	3	
14.	Создание игры Шутер	8	4	4	
15.	Доработка и презентация проекта	2	1	1	
Модуль 6. Публикация и распространение ПО		10			
16.	Сборка проекта в приложение	2	1	1	Задания в среде программирования с расширением учебной платформы Алгоритмики
17.	Повторение. Введение в Git	2	1	1	
18.	Создание игры Пинг-понг	4	2	2	

19.	Выпускной	2	1	1	Задания на платформе
-----	-----------	---	---	---	----------------------

2.2. Содержание учебного (тематического) плана

Модуль 1. Структуры данных

1.1. Повторение. Обработка исключений

Функции ввода и вывода, переменные и типы данных. Алгоритмические конструкции - следование, ветвление и цикл. Модули Random и Time. Обработка исключений, операторы try, except.

Решение задач на повторение на платформе.

1.2. Повторение. Списки

Списки. Методы работы со списком: добавление, удаление, вставка, сортировка, подсчет вхождений. Типы данных, которые может содержать список. Перебор элементов списка с помощью цикла for. Определение понятия Строка. Аналогии между методами работы со списками и со строками.

1.3. Словари

Определение понятия Словарь. Добавление в словарь пары “ключ-значение”. Методы работы с ключами и значениями словаря. Поиск ключа в словаре. Типы данных ключей и значений.

1.4. Вложенные структуры данных

Правила создания лаконичного кода. Вложенные структуры. Применение методов к вложенным структурам. Обращение к объектам внутри вложенных структур с помощью индексов. Решение задач повышенной сложности.

Модуль 2. Разработка оконных приложений

2.1. Классы. Введение в PyQt

Определение Класа. Связь классов и объектов. Конструкторы, поля и методы в классах. Приложения и виджеты. Расположение виджетов. Обработка событий средствами PyQt.

2.2. Проектирование интерфейса

Наследование. Суперклассы, классы-наследники, их методы. Виджеты QRadioButton, QMessageBox и др.

2.3. Последовательное создание профессионального (компьютерного) приложения Memory Card

Основные элементы приложения: окно, текст, кнопки, флажки, переключатели, списки, меню. Проектирование интерфейса приложения с помощью лейаутов и виджетов.

Модуль 3. Работа с файлами

3.1. Основы работы с файлами

Атрибуты доступа к файлам. Оператор with. Чтение информации из файла. Строки и классы для считывания информации из файла.

3.2. Создание оконного приложения «Умные заметки» с поиском записей по тегам

Json-файлы. Запись и чтение данных из json-файлов. Глобальные и локальные переменные. Программирование корректного отображения данных из файла.

Создание приложения для работы с заметками с возможностью создавать, удалять и редактировать заметки, задавать заметкам теги и выполнять поиск по ним.

Модуль 4. Автоматическая обработка изображений

4.1. Основы обработки изображений

Библиотека Python PIL как инструмент обработки изображений. Модули Image и ImageFilter библиотеки PIL. Параметры изображения Image, их изменение с помощью PIL.

4.2. Создание приложения Easy Editor для обработки фотографий

Возможности и методы обработки изображений библиотекой PIL. Локальные и глобальные переменные. Программирование функций работы с файлами и интерфейсом. Массовая обработка изображений.

Создание приложения-фоторедактора Easy Editor (загрузка фото с компьютера, наложение ч/б фильтра, наведение резкости и поворот изображения).

Модуль 5. Продвинутая разработка игр на PyGame

5.1. Основы создания игр

Обработка нажатия клавиши с помощью PyGame. Создание окна игры и спрайтов. Игровой цикл. Событие завершения игры.

5.2. Создание игры Лабиринт

Класс Sprite. Метод `super().__init__()`. Создание поля, метода и экземпляров собственного класса. Условия победы и поражения. Обработка внутриигровых событий и внешней среды.

5.3. Создание игры Шутер

Создание класса наследника. Задание спрайтов, управляемых с клавиатуры. Задание аудио и визуального фона игры. Создание группы спрайтов и управление ей. Тест-кейсы и тестирование.

5.4. Доработка и презентация проекта

Доработка проекта. Презентация проекта. Приемы успешных публичных выступлений. Google-презентации.

Модуль 6. Бонусный модуль — портфолио разработчика

6.1. Сборка проекта в приложение

Расширения. Система управления пакетов `pip`. Модуль `Pyinstaller`. Упаковка игры в файл с расширением `.exe` (`.pkg`).

6.2. Повторение. Введение в Git

Система контроля версий. Сервис `GitHub`. Понятия репозиторий, ветка, коммит. Создание репозитория с проектом на `PyQt`.

6.3. Создание игры Пинг-Понг

Планирование проекта с помощью `mind map`. Реализация проекта по плану. Выбор лицензии на `GitHub` и заполнение файла `ReadMe`.

6.4. Выпускной

Обобщение и систематизация пройденного материала в игровой форме (командный хакатон по материалам курса).

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 и в среде программирования с расширением учебной платформы Алгоритмики , реализация проектов, самоконтроль ученика;
- **в конце каждого модуля:** презентации финальных проектов модуля и их оценка.

Для контроля сформированности результатов освоения программы с помощью выполнения заданий на платформе используются следующие механики:

- классический тест (выбор одного или нескольких правильных ответов),
- заполнение пропусков,
- классификация,
- сопоставление,
- сортировка.

На курсе возможны следующие **формы деятельности** учащихся: учебная дискуссия, занятие-проект, практическое занятие, конференция. При решении одной задачи ученики могут работать сначала в малой группе, затем индивидуально, а при проверке задачи — в паре с учителем.

Задания на платформе с автопроверкой (все задания на тему базовых понятий проверяются автоматически системой) направлены на формирование навыков написания программ на языке Python и умения читать код, вносить в него изменения (исправления) и дополнять. Кроме того, предусмотрены ревью кода (выборочная оценка кода учеников на соблюдение стандартов его написания), конференции и круглые столы, на которых возможна взаимопроверка, чек-листы для преподавателей и учеников.

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;
- проектор, подключенный к компьютеру для учителя, выводящий изображение на экран;
- распечатанный список логинов и паролей учеников для доступа на платформу;
- платформа «Алгоритмика» с интеграцией PyGame, открывается через браузер Google Chrome;
- Среда программирования VS Code с расширением учебной платформы Алгоритмики

- 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 и в среде программирования с расширением учебной платформы Алгоритмики ;
- справочный теоретический материал, размещенный на платформе;
- бонусные задания на платформе;
- дополнительные проектные задания;
- задачи со свободным решением для реализации в среде разработки на базе платформы «Алгоритмика».

Календарный учебный график

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Количество часов	Тема занятия	Место проведения	Форма контроля
1				Интерактивное занятие	2	Повторение. Обработка исключений		Задания на платформе
2				Интерактивное занятие	2	Повторение. Списки		Задания на платформе
3				Интерактивное занятие	2	Словари		Задания на платформе
4				Интерактивное занятие	2	Вложенные структуры данных		Задания на платформе
5				Интерактивное занятие	2	Классы. Введение в PyQt		Задания в среде программирования с расширением учебной платформы Алгоритмики
6				Интерактивное занятие	2	Проектирование интерфейса		Задания в среде программирования с расширением учебной платформы Алгоритмики
7				Интерактивное занятие	2	Последовательное создание оконного приложения Memory Card ч.1		Задания в среде программирования с расширением учебной платформы Алгоритмики
8				Интерактивное занятие	2	Последовательное создание оконного приложения Memory Card ч.2		Задания в среде программирования с расширением учебной платформы Алгоритмики
9				Интерактивное занятие	2	Последовательное создание оконного приложения Memory Card ч.3		Задания в среде программирования с расширением учебной платформы Алгоритмики

10				Интерактивное занятие	2	Последовательное создание профессионального оконного Memory Card ч.4		Задания в среде программирования с расширением учебной платформы Алгоритмики
11				Интерактивное занятие	2	Основы работы с файлами		Задания в среде программирования с расширением учебной платформы Алгоритмики
12				Интерактивное занятие	2	Создание оконного приложения «Умные заметки» с поиском записей по тегам ч.1		Задания в среде программирования с расширением учебной платформы Алгоритмики
13				Интерактивное занятие	2	Создание оконного приложения «Умные заметки» с поиском записей по тегам ч.2		Задания в среде программирования с расширением учебной платформы Алгоритмики
14				Интерактивное занятие	2	Создание оконного приложения «Умные заметки» с поиском записей по тегам ч.3		Задания в среде программирования с расширением учебной платформы Алгоритмики
15				Интерактивное занятие	2	Основы обработки изображений		Задания в среде программирования с расширением учебной платформы Алгоритмики
16				Интерактивное занятие	2	Создание приложения Easy Edior для обработки фотографий ч.1		Задания в среде программирования с расширением учебной платформы Алгоритмики
17				Интерактивное занятие	2	Создание приложения Easy Edior для обработки фотографий ч.2		Задания в среде программирования с расширением учебной платформы Алгоритмики
18				Интерактивное занятие	2	Создание приложения Easy		Задания в среде программирования с расширением

						Edior для обработки фотографий ч.3		учебной платформы Алгоритмики
19				Интерактивное занятие	2	Основы создания игр		Задания в среде программирования с расширением учебной платформы Алгоритмики
20				Интерактивное занятие	2	Создание игры Лабиринт ч.1		Задания в среде программирования с расширением учебной платформы Алгоритмики
21				Интерактивное занятие	2	Создание игры Лабиринт ч.2		Задания в среде программирования с расширением учебной платформы Алгоритмики
22				Интерактивное занятие	2	Создание игры Лабиринт ч.3		Задания в среде программирования с расширением учебной платформы Алгоритмики
23				Интерактивное занятие	2	Создание игры Шутер ч.1		Задания в среде программирования с расширением учебной платформы Алгоритмики
24				Интерактивное занятие	2	Создание игры Шутер ч.2		Задания в среде программирования с расширением учебной платформы Алгоритмики
25				Интерактивное занятие	2	Создание игры Шутер ч.3		Задания в среде программирования с расширением учебной платформы Алгоритмики
26				Интерактивное занятие	2	Создание игры Шутер ч.4		Задания в среде программирования с расширением учебной платформы Алгоритмики

27				Интерактивное занятие	2	Доработка и презентация проекта		Задания в среде программирования с расширением учебной платформы Алгоритмики
28				Интерактивное занятие	2	Сборка проекта в приложение		Задания в среде программирования с расширением учебной платформы Алгоритмики
29				Интерактивное занятие	2	Повторение. Введение в Git		Задания в среде программирования с расширением учебной платформы Алгоритмики
30				Интерактивное занятие	2	Создание игры Пинг-понг ч.1		Задания в среде программирования с расширением учебной платформы Алгоритмики
31				Интерактивное занятие	2	Создание игры Пинг-понг ч.2		Задания в среде программирования с расширением учебной платформы Алгоритмики
32				Интерактивное занятие	2	Выпускной		Задания на платформе

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. Эльконин Д.Б. Психология развития: Учеб. пособие для студ. высш. учеб. заведений. М., 2001.
7. Асмолов А. Г., Бурменская Г. В., Володарская И. А. и др. Формирование универсальных учебных действий в основной школе: от действия к мысли. Система заданий. / Под ред. Асмолова А. Г. — 2-е изд. — М.: Просвещение, 2011. — 159 с.
8. Брайсон Пэйн. Python для детей и родителей. — ООО «Эксмо», 2017. — 352 с.
9. Вордерман Кэрол, Джон Вудкок, Шон Макаманус. Программирование для детей. Иллюстрированное руководство для детей; пер. с англ. С. Ломакина. — 4-е изд. М.: Манн, Иванов и Фербер, 2018, — 224 с.
10. Курс: «Python для взрослых и детей» [Электронный ресурс]. — Режим доступа: <https://stepik.org/course/4401>.
11. Фридман Л.М. Как научиться решать задачи. — М.: Просвещение, 1989.
12. А.А. Вербицкий, О.Г. Ларионова: Личностный и компетентностный

подходы в образовании: проблемы интеграции. Монография. — М.: Издательство «Логос», 2017. — 336 с.

13. 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.

14. Салман Хан. Весь мир - школа. Преобразованное образование/ The One World Schoolhouse: Education Reimagined. [текст]- Арт-транзит, Классика-XXI, 2015. – 176 с.

15. Самылкина Н.Н. Современные средства оценивания результатов обучения. М.: БИНОМ, 2007. – 172 с.

16. М.Э. Абрамян Programming Taskbook Электронный задачник по программированию Версия 4.6 Ростов-на-Дону 2007.

17. Босова Л.Л. Как учат программированию в XXI веке: отечественный и зарубежный опыт обучения программированию в школе. // Информатика в школе. 2018. №6. С. 3-11.