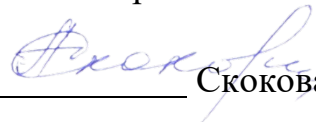


**ООО «Алгоритмика-Пермь»**

УТВЕРЖДАЮ

Директор ООО «Алгоритмика-Пермь».



  
Скокова А.В.

**Дополнительная общеобразовательная программа –**

**Дополнительная общеразвивающая программа**

***«Курс программирования Python Старт»***

***1 год обучения***

Направленность: техническая

Уровень программы: ознакомительный

Возраст учащихся: 12–13 лет

Срок реализации: 1 учебный год (64 академических часа)

Авторы-составители:

Скокова А.В.

г. Пермь, 2024 год

## 1. Пояснительная записка

### 1.1. Направленность и уровень Программы

Программирование уже изменило повседневную жизнь. Оно поменяло наши привычки искать информацию, подходы к заботе о здоровье, прослушивание музыки, общение с друзьями. Системные занятия программированием формируют комплекс навыков, которые можно использовать в повседневной жизни. Среди них умение непрерывно учиться новому, использование передовых технологий, работа в команде, быть гибкими и адаптироваться, внимательность, самостоятельность и инициативность, способность к рефлексии и высокий уровень критического мышления.

Курс программирования «Python Старт» 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 пройти через различные этапы работы в большой IT-компании. Ученик проходит путь от собеседования и начала карьеры стажера-разработчика (intern python developer) до ведущего разработчика (senior python developer). Данные задачи не только способствуют формированию предметных компетенций, но еще и формируют взгляд ученика на мир, раскрывают значение soft skills, учат грамотному использованию компьютера и готовят к жизни в цифровом обществе.

В рамках курса учащиеся освоят базовые принципы программирования на языке Python через создание собственных проектов и решение задач разной сложности (от исправления ошибки в чужой программе до запуска beta-версии своего продукта).

Особенность курса проявляется в том, что изучив необходимый теоретический материал, учащимся предлагается самостоятельно решить практическую задачу, представленную в виде адаптированного кейса из реальной жизни. Для этого, учащиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают ее с помощью формального языка. Полученные знания учащиеся применяют при создании собственных проектов, которые защищаются 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 основных навыков программирования на языке Python;
- формировании и развитии навыков работы с информацией в условиях модернизации образования;
- формировании навыков предоставления и получения обратной связи;
- формировании понимания информационной безопасности;
- формировании навыков управления проектами;
- развитию цифровой и функциональной грамотности;
- развитию алгоритмического и креативного мышления, творчества через создание собственных проектов с использованием цифровых ресурсов;
- необходимости расширения кругозора школьника.

**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 как программисты, а не просто решать стандартный набор задач. Более того, проекты, выполняемые на уроках, охватывают разные области интересов, в том числе школьные предметы. 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 инструменты профессиональной разработки для реализации собственных проектов. Для этого сначала ученик получает базовые знания (алгоритмы, структурное программирование, использование библиотек в процедурном стиле), затем учится навыкам проектной деятельности, после чего реализует проекты в области своих увлечений. При этом базовая часть также подразумевает возможность выразить свои интересы, благодаря большому количеству задач разного уровня. Проекты выполняются преимущественно индивидуально, чтобы дети сначала научились работать над ними, затем уже — работать в группе. Эволюция компьютеров и

программного обеспечения 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 и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих 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 принципами объектно-ориентированного программирования.
4. Сформировать навык правильного оформления кода.
5. Сформировать навык использования итеративного подхода при решении различных задач.
6. Формировать навык решения задач и разработки проектов с помощью Python.
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, креативное и пространственное 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. Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.

7. Совершенствовать диалогическую 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.

#### **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.

*Формы 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.

*Занятие состоит из следующих блоков:*

- Повторение и определение целей на урок (5 минут).
- Изучение нового материала (20 минут).
- Практика (15 минут).
- Разминка (10 минут).
- Изучение нового материала (10 минут).
- Практика (20 минут).
- Рефлексия: подведение итогов занятия (10 минут).

## 1.5. Срок реализации Программы

Курс программирования «Python Start» (1 год обучения) для обучающихся 12–13 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
<b>Предметные навыки</b>		
Формирование знаний об основах программирования, в т. ч. алгоритмических конструкциях, логических значениях и операциях.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе с постепенным увеличением сложности.	Ученик оперирует базовыми понятиями основ программирования: алгоритм, цикл, условие, функция, переменная, структуры данных и т. д.
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.
Следование стандарту при оформлении кода.	Изучение теоретического материала совместно с	Ученик оформляет код в соответствии с общепринятыми

	учителем, выполнение заданий на платформе.	стандартами и использует комментарии.
Умение тестировать программу и находить в ней ошибки.	Выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно определяет и исправляет ошибки в своем и чужом коде.
Умение использовать чужой код при разработке: анализировать и изменять.	Выполнение заданий на платформе.	Ученик самостоятельно может внести изменения в готовый код и использовать фрагмент готового кода в процессе реализации собственной программы.
Использование итерационного подхода.	Выполнение заданий на платформе.	Ученик пишет программы, разделяя код на отдельные части.

<b>Личностные навыки</b>		
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	Использование программирования как способа показать преимущества работы в команде.	Ученик не боится просить помощь и сам пытается помогать одноклассникам и учителю.

<p>находить выходы из спорных ситуаций.</p>		
<p>Овладение начальными навыками адаптации в динамично изменяющемся и развивающемся мире.</p>	<p>Обсуждение научно-популярных материалов, демонстрирующих необходимость не просто овладения предметными навыками, но и развития умения учиться.</p>	<p>При работе над проектом ученик сначала пытается найти информацию в Интернете или теоретической справке на платформе и только при неудаче задает вопрос учителю.</p>

<p align="center"><b>Мегапредметные навыки</b></p>		
<p>Развитие и формирование учебных действий.</p>	<p>Создание благоприятных условий для участия в диалоге, в коллективном обсуждении. Строится продуктивное взаимопонимание со сверстниками и взрослыми в процессе коллективной деятельности.</p>	<p>Ученик легко общается, не боится просить помощь или оказать ее другим. Ученик способен кооперироваться, чтобы достичь цели.</p>
<p>Умение презентовать свою работу.</p>	<p>Презентация индивидуальных проектов. Учитель и другие ученики</p>	<p>Ученик во время презентации своих проектов пользуется вниманием аудитории.</p>

	дают обратную связь. Учитель также дает советы, каким образом это лучше делать.	
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	
<b>Модуль 1. Основы языка</b>		<b>8</b>			
1	Введение в Python. Функции ввода-вывода	2	1	1	Задания на платформе
2.	Переменные. Численные типы данных	2	1	1	Задания на платформе
3.	Строки	2	1	1	Задания на платформе
4.	Вложенные конструкции	2	1	1	Задания на платформе
<b>Модуль 2. Управляющие конструкции</b>		<b>10</b>			

5.	Логический тип данных. Условный оператор	2	1	1	Задания на платформе
6.	Вложенный условный оператор и условный оператор нескольких ветвей	2	1	1	Задания на платформе
7.	Цикл while. Цикл со счетчиком.	4	2	2	Задания на платформе
8.	Вложенные алгоритмические конструкции (цикл в цикл, условный оператор в цикл и др.)	2	1	1	Задания на платформе
<b>Модуль 3. Функции и модули</b>		<b>8</b>			
9	Создание функций. Локальная и глобальная области видимости	2	1	1	Задания на платформе
10.	Использование одной функции внутри другой	2	1	1	Задания на платформе
11.	Модули. Использование встроенных модулей стандартной библиотеки	2	1	1	Задания на платформе
12	Создание модулей. Подключение нескольких модулей	2	1	1	Задания на платформе
<b>Модуль 4. Модуль Turtle. Математика для разработчика</b>		<b>8</b>			
13.	Графический исполнитель Turtle. Математика перемещения исполнителя	2	1	1	Задания на платформе
14.	Условный оператор. Рисование простых геометрических фигур	2	1	1	Задания на платформе
15.	Цикл. Рисование многоугольников	2	1	1	Задания на платформе
16.	Проектное занятие	2	1	1	Задания на платформе
<b>Модуль 5. Объектно-ориентированное программирование</b>		<b>12</b>			
17.	Объекты, их поля и методы	2	1	1	Задания на платформе
18.	Обработка событий мыши и клавиатуры	6	3	3	Задания на платформе
19.	Классы. Конструктор класса	2	1	1	Задания на платформе
20.	Наследование (от готового или своего класса)	2	1	1	Задания на платформе
<b>Модуль 6. Основы разработки игр на PyGame</b>		<b>16</b>			

21.	Основы разработки игр. Спрайт, игровой цикл	2	1	1	Задания на платформе
22.	Списки и методы работы с ними. Перебор списка в цикле for	2	1	1	Задания на платформе
23.	Обработка внутриигровых событий	4	2	2	Задания на платформе
24.	Игровая физика	8	4	4	Задания на платформе
25.	<b>Выпускной</b>	2	1	1	Задания на платформе

## 2.2. Содержание учебного (тематического) плана

### Модуль 1. Основы языка

#### 1.1. Введение в Python. Функции ввода-вывода

*Язык программирования Python. Понятия «язык программирования», «алгоритм», «программа». Знакомство с синтаксисом языка. Функция ввода input(). Функция вывода print(). Необходимость вывода данных.*

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

#### 1.2. Переменные. Численные типы данных

*Данные. Типы данных. Переменная. Числовой тип данных. Оператор присваивания. Приоритет операций. Выражение.*

*Разработка программ, осуществляющих обработку переменных числового типа.*

#### 1.3. Строки

*Строковый тип данных. Операции со строками и числами. Изменение типов данных с помощью функций int() и str().*

*Разработка программ, осуществляющих обработку строковых переменных (сумма строк, повтор строки n-раз, вычисление длины строки, получение символа строки по номеру, замена подстроки на другую подстроку, разбиение строки по разделителю).*

#### 1.4. Вложенные конструкции

### Модуль 2. Управляющие конструкции

## **2.1. Логический тип данных. Условный оператор**

*Логический тип данных. Утверждение. Условие. Условный оператор. Синтаксис условного оператора. Операторы сравнения.*

*Разработка программ, содержащих условный оператор.*

## **2.2. Вложенный условный оператор и условный оператор нескольких ветвей**

*Вложенный условный оператор. Уровни вложенности. Условный оператор нескольких ветвей.*

*Разработка программ, содержащих вложенный условный оператор.*

## **2.3. Цикл while. Цикл со счетчиком**

*Цикл. Необходимость применения циклов. Типы циклов. Демонстрация синтаксиса цикла while(). Разбор примеров программ с использованием цикла. Цикл со счетчиком. Цикл с накоплением результата вычислений в переменной.*

*Разработка программ, содержащих циклы различных типов.*

## **2.4. Вложенные алгоритмические конструкции (цикл в цикл, условный оператор в цикл и др.)**

*Вложенные алгоритмические конструкции. Цикл в цикле. Цикл с вложенными условными операторами. Целочисленное деление.*

*Разработка программ, осуществляющих обработку данных с помощью вложенных алгоритмических конструкций.*

## **Модуль 3. Функции и модули**

### **3.1. Создание функций. Локальная и глобальная области видимости**

*Функция. Локальная и глобальная функции. Оператор def. Параметры функции. Значение функции. Вызов функции. Синтаксис создания функций.*

*Разработка программ, осуществляющих обработку данных с помощью функций.*

### **3.2. Использование одной функции внутри другой**

*Использование одной функции внутри другой.*

*Разработка программ, содержащих функцию внутри другой функции.*

### **3.3. Модули. Использование встроенных модулей стандартной библиотеки**

*Модуль. Встроенные модули стандартной библиотеки. Модуль random.*

*Разработка программ, осуществляющих обработку данных с помощью модулей, встроенных в стандартную библиотеку.*

### **3.4. Создание модулей. Подключение нескольких модулей**

*Создание модуля. Подключение модуля.*

*Разработка программ, осуществляющих обработку данных с помощью дополнительных модулей, не входящих в стандартную библиотеку.*

## **Модуль 4. Модуль Turtle. Математика для разработчика**

### **4.1. Графический исполнитель Turtle. Математика перемещения исполнителя**

*Исполнитель. Графический исполнитель. Пиксель. Координаты. Знакомство с модулем Turtle: рисование линий, поворот ширина и цвет пера, перемещение пера в указанную точку. Математика перемещения исполнителя.*

*Разработка программ, управляющих исполнителем.*

### **4.2. Условный оператор. Рисование простых геометрических фигур**

*Условный оператор в модуле Turtle. Рисование простых геометрических фигур.*

*Разработка программ, осуществляющих рисование простых геометрических фигур с использованием условного оператора.*

### **4.3. Цикл. Рисование многоугольников**

*Циклы в модуле Turtle. Рисование многоугольников.*

*Разработка программ, осуществляющих рисование многоугольников.*

### **4.4. Проектное занятие**

## **Модуль 5. Объектно-ориентированное программирование**

### **5.1. Объекты, их поля и методы**

*Объекты. Поля и методы объектов. Создание объектов.*

*Разработка программ, осуществляющих обработку данных с помощью передачи объекта в функцию.*

## **5.2. Обработка событий мыши и клавиатуры**

*События. Обработка взаимодействия пользователя и программы. Обработка событий мыши и клавиатуры.*

*Разработка программ, осуществляющих обработку событий.*

## **5.3. Классы. Конструктор класса**

*Классы. Конструктор класса. Создание нестандартного объекта и описание его свойств и методов.*

*Разработка программ, осуществляющих обработку классов.*

## **5.4. Наследование (от готового или своего класса)**

*Наследование. Класс-наследник. Использование готовых свойств и методов объекта. Добавление новых методов.*

*Разработка программ, использующих наследование.*

## **Модуль 6. Основы разработки игр на PyGame**

### **6.1. Основы разработки игр. Спрайт, игровой цикл**

*Спрайт. Игровой цикл: обработка событий, обновление игры, прорисовка, подсчет времени. Библиотека play. Создание спрайтов разных типов и с разными свойствами. Расположение спрайта.*

*Разработка программ для создания спрайтов разных типов и управления ими.*

### **6.2. Списки и методы работы с ними. Перебор списка в цикле for**

*Списки. Методы работы со списками. Перебор списков в цикле for.*

*Разработка программ, осуществляющих обработку списков.*

### **6.3. Обработка внутриигровых событий**

*Внутриигровые события. Обработка внутриигровых событий.*

*Разработка программ, осуществляющих обработку внутриигровых событий.*

### **6.4. Игровая физика**

*Игровая физика. Работа со спрайтами как с физическими объектами. Подключение физики. Применение инструментов при создании игр.*

*Разработка игр с использованием встроенной симуляции физики.*

## **6.7. 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;
- **в конце каждого модуля:** проведение мини-хакатонов и 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;
- 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;
- справочный теоретический материал, размещенный на платформе;
- бонусные задания на платформе;
- 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	Введение в Python. Функции ввода-вывода.		Задания на платформе
2				Интерактивное занятие	2	Переменные. Численные типы данных.		Задания на платформе
3				Интерактивное занятие	2	Строки		Задания на платформе
4				Интерактивное занятие	2	Вложенные конструкции		Задания на платформе
5				Интерактивное занятие	2	Логический тип данных. Условный оператор		Задания на платформе
6				Интерактивное занятие	2	Вложенный условный оператор и условный оператор нескольких ветвей		Задания на платформе
7				Интерактивное занятие	2	Цикл while. Цикл со счетчиком		Задания на платформе
8				Интерактивное занятие	2	Цикл while. Цикл со счетчиком		Задания на платформе
9				Интерактивное занятие	2	Вложенные алгоритмические конструкции (цикл в цикл, условный оператор в цикл и др.)		Задания на платформе
10				Интерактивное занятие	2	Создание функций. Локальная и глобальная области видимости		Задания на платформе

11				Интерактивное занятие	2	Использование одной функции внутри другой		Задания на платформе
12				Интерактивное занятие	2	Модули. Использование встроенных модулей стандартной библиотеки		Задания на платформе
13				Интерактивное занятие	2	Создание модулей. Подключение нескольких модулей		Задания на платформе
14				Интерактивное занятие	2	Графический исполнитель Turtle. Математика перемещения исполнителя		Задания на платформе
15				Интерактивное занятие	2	Условный оператор. Рисование простых геометрических фигур		Задания на платформе
16				Интерактивное занятие	2	Цикл. Рисование многоугольников		Задания на платформе
17				Интерактивное занятие	2	Проектное занятие		Задания на платформе
18				Интерактивное занятие	2	Объекты, их поля и методы		Задания на платформе
19				Интерактивное занятие	2	Обработка событий мыши и клавиатуры		Задания на платформе
20				Интерактивное занятие	2	Обработка событий мыши и клавиатуры		Задания на платформе

21				Интерактивное занятие	2	Обработка событий мыши и клавиатуры		Задания на платформе
22				Интерактивное занятие	2	Классы. Конструктор класса		Задания на платформе
23				Интерактивное занятие	2	Наследование (от готового или своего класса)		Задания на платформе
24				Интерактивное занятие	2	Основы разработки игр. Спрайт, игровой цикл		Задания на платформе
25				Интерактивное занятие	2	Списки и методы работы с ними. Перебор списка в цикле for.		Задания на платформе
26				Интерактивное занятие	2	Обработка внутриигровых событий		Задания на платформе
27				Интерактивное занятие	2	Обработка внутриигровых событий		Задания на платформе
28				Интерактивное занятие	2	Игровая физика		Задания на платформе
29				Интерактивное занятие	2	Игровая физика		Задания на платформе
30				Интерактивное занятие	2	Игровая физика		Задания на платформе
31				Интерактивное занятие	2	Игровая физика		Задания на платформе
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](http://минобрнауки.рф/документы/543).
4. Федеральный закон «Об образовании в РФ» [Электронный ресурс] Режим доступа: [consultant.ru/document/cons\\_doc\\_LAW\\_140174/](http://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.