

УТВЕРЖДАЮ  
Ректор  
ФГБОУ ВО «РГУПС»



В.Н. Верескун  
М.П.

УТВЕРЖДАЮ  
Президент  
БФ «Система»



О.П. Косаченко  
М.П.

УТВЕРЖДАЮ  
Генеральный директор  
ООО «Алгоритмика»



А.С. Лобанов  
М.П.

## ДОПОЛНИТЕЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ

Возраст обучающихся: 14–17 лет  
Уровень: модульная образовательная программа  
«Курс программирования Python Pro»

Авторы-составители:

Мария Липчанская  
Ирина Курзенко

Москва, Ростов  
2021 г.

## СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
1. НОРМАТИВНО-ПРАВОВАЯ БАЗА .....	5
2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ.....	6
3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ.....	8
4. УЧЕБНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН .....	16
5. СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА.....	21
6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ .....	26

Приложение 1. Список литературы

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данная общеобразовательная общеразвивающая программа дополнительного образования детей имеет техническую направленность и реализуется в рамках благотворительной программы «Лифт в будущее. Школа».

Программа предполагает дополнительное образование детей в области программирования на языке Python.

Организация образовательной деятельности в рамках программы основана на модульном принципе представления содержания образовательной программы, построения учебных планов и использовании современных образовательных технологий, в сетевом взаимодействии, с использованием ресурсов нескольких организаций.

В рамках программы обучающиеся освоят 5 модулей:

1 модуль — «Типы данных»;

2 модуль — «Алгоритмы»;

3 модуль — «Способы организации кода»;

4 модуль — «Структуры данных»;

5 модуль — «Объектно-ориентированное программирование».

**Содержание курса** программирования Python Pro для обучающихся 14–17 лет рассчитано на обучение в течение 6 месяцев (22 занятия — 44 академических часа). Занятия проходят 1 раз в неделю и длятся 2 академических часа. Предусмотрены необязательные домашние задания. Отбор на курс предусмотрен только по возрасту.

Программа курса 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 Soft skills: умение работать в команде, планирование, креативность, критическое мышление, навыки проведения презентаций и публичных выступлений.

В отличие от других курсов программирования Python Pro учит ребенка думать, как программист, а не просто писать код. Более того, проекты, выполняемые на уроках, охватывают различные области интересов, в том числе школьные предметы.

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

**Особенность курса** заключается в том, что детям не дают образец решения задачи.

Ученики сами пытаются составить последовательность действий, ведущую к цели, и запрограммировать ее. Полученные знания ребенок применяет при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями.

**Теоретической основой данной программы** являются:

Компетентностный подход: обучение в системе, развивающей предметные способности. Развитие каждой способности происходит в три шага: формирование учебной мотивации; формирование предметных знаний, умений и навыков; формирование личностных изменений, связанных с усвоением новых знаний (Хуторской А. В. и др.)

Личностный подход: обучение в условиях развития каждого учащегося как личности (Вербцкий А. А. и др.).

## 1. НОРМАТИВНО-ПРАВОВАЯ БАЗА

Программа разработана в соответствии с нормативно-правовыми документами:

1. Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

2. Приказ Минпросвещения России от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

3. Постановление Главного государственного санитарного врача Российской Федерации от 04.07.2014 № 41 Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

## 2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель всего курса — научить ребенка использовать язык программирования Python как инструмент 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 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 решения задач и разработки проектов с помощью Python.
8. Формировать навык разработки игр с помощью языка программирования Python.
9. 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 компетенций, но еще и формируют мировоззрение школьника, раскрывают значение Soft skills, учат грамотному использованию компьютера и готовят к жизни в цифровом обществе.

### 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Достижение цели и задач образовательной программы предполагает получение следующих результатов:

Планируемые результаты	Способ достижения	Критерий достижения образовательного результата
<b>Предметные навыки</b>		
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	Ученик для выполнения проекта без подсказки учителя выбирает оптимальную из известных ему структуры данных.



	и доказывают, почему это лучшее решение.	
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.
<b>Личностные навыки</b>		
Освоение социальной роли обучающегося и	Демонстрация связи между способностью	За отведенное время ученик пытается не

<p>формирование личностного смысла учения.</p>	<p>выполнить интересную задачу и наличием/отсутствием соответствующих знаний.</p>	<p>только выполнить базовые уровни, но и приступить к бонусным.</p>
<p>Развитие навыков сотрудничества со взрослыми и сверстниками и умения находить выходы из спорных ситуаций.</p>	<p>Использование программирования как способа показать преимущества работы в команде.</p>	<p>Ученик не боится просить помощь и сам пытается помогать одноклассникам и учителю.</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>Умение аргументировать свою позицию.</p>	<p>Выступление во время защиты проектов, ответы на вопросы.</p>	<p>Ученик обосновывает подход к решению проблемы, используя цепочки аргументов. Помнит, как использовались логические связи между</p>

		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.

<p>Понимание того, что компьютер — уникальный инструмент, с помощью которого можно добиться многого.</p>	<p>Акцентирование внимания на том, что с помощью компьютера делаются вещи, которые без него делались бы месяцами (или были бы невозможны вообще).</p>	<p>Ученик в соответствующих тестовых заданиях ставит компьютеру самую высокую оценку полезности.</p>
<p><b>Метапредметные 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 с другими, чтобы достичь цели.</p>
<p>Умение презентовать свою работу.</p>	<p>Презентация индивидуальных проектов. Учитель и другие ученики дают обратную связь. Учитель также дает</p>	<p>Ученик во время презентации своих проектов пользуется вниманием аудитории.</p>

	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.
Умение управлять своим временем: устанавливать приоритеты, 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 о том, 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.

Механизмы отслеживания результатов:

1. Задания на платформе с автопроверкой (все задания на тему базовых понятий проверяются автоматически системой).
2. Ревью кода (выборочная оценка кода учеников на соблюдение стандартов его написания).
3. Конференции и «круглые столы», на которых возможна взаимопроверка.

4. Чек-листы для преподавателей, других учеников (большинство метанавыков оценивается через заполнение преподавателями и учениками анкет).
5. Анкета родителей (для оценки появления и развития установок).

Учебно-дидактическое обеспечение:

1. Поурочные методические указания.
2. Тематические презентации.
3. Персональные компьютеры по числу учащихся в группе.
4. Задания на платформе <https://lk.bf.sistema.ru> в личных ученических аккаунтах.
5. Компьютер, доска, проектор для учителя.

#### 4. УЧЕБНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

В рамках личностного подхода, реализуемого в Python Start, в учебном тематическом плане (УТП) указана примерная продолжительность каждого модуля, поскольку преподаватель может ускорять или замедлять темп урока для каждой группы.

Например, если преподаватель видит, что группа не усвоила тему за отведенное время, то он имеет право повторить урок.

№	Наименование раздела	Количество академических часов			Форма аттестации
		всего	теория	практика	
<i>Модуль 1. Типы данных</i>					
1	Введение в язык программирования	2	1	1	Тестирование, задания на платформе с автопроверкой
2	Переменные	2	1	1	Тестирование, задания на платформе с автопроверкой
3	Строки	2	1	1	Тестирование, задания на платформе с автопроверкой
4	Практикум по решению задач	2	1	1	Задания на платформе с автопроверкой, создание программ в Visual Studio Code



	<i>Модуль 2. Алгоритмы</i>				
5	Условный оператор	2	1	1	Задания на платформе с автопроверкой, создание программ в Visual Studio Code
6	Вложенный условный оператор	2	1	1	Задания на платформе с автопроверкой, создание программ в Visual Studio Code
7	Циклы	4	2	2	Задания на платформе с автопроверкой, создание программ в Visual Studio Code
	<i>Модуль 3. Способы организации кода</i>				
8	Функции	4	2	2	Задания на платформе с автопроверкой, создание программ в Visual Studio Code
9	Модули	2	1	1	Задания на платформе с автопроверкой, создание программ в Visual Studio Code
10	Командный хакатон	2	1	1	Создание программ в Visual Studio Code в

					соответствии с техническим заданием, самостоятельное тестирование программы, ревью программы
<i>Модуль 4. Структуры данных</i>					
11	Списки и кортежи	2	1	1	Тестирование, создание программ в Visual Studio Code
12	Словари и множества	2	1	1	Тестирование, создание программ в Visual Studio Code
13	Вложенные структуры данных	2	1	1	Тестирование, создание программ в Visual Studio Code
14	Практикум по решению задач	2	1	1	Создание программ в Visual Studio Code в соответствии с техническим заданием, самостоятельное тестирование программы, ревью программы

<i>Модуль 5. Объектно-ориентированное программирование</i>					
15	Классы	2	1	1	Тестирование, создание программ в Visual Studio Code
16	Наследования	2	1	1	Тестирование, создание программ в Visual Studio Code
17	Игра Fast Clicker	4	2	2	Создание программ в Visual Studio Code в соответствии с техническим заданием, самостоятельное тестирование программы, ревью программы одноклассником, ревью программы учителем
18	Выполнение проектов с помощью различных библиотек (сборников подпрограмм или объектов, используемых для разработки	4	2	2	Задания на платформе с автопроверкой, создание программ в Visual Studio Code в соответствии с техническим заданием, самостоятельное тестирование

	программного обеспечения)				программы, ревью программы одногогруппником, ревью программы учителем
--	------------------------------	--	--	--	---

## 5. СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

### Тема 1.1. Знакомство с языком

#### Теория

Язык программирования Python. Знакомство с синтаксисом языка. Переменная. Значение переменной. Оператор присваивания. Функция. Вызов функции. Параметры. Функция input().

#### Практика

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

### Тема 1.2. Переменные

#### Теория

Понятия «данные», «переменные», «оператор присваивания». Необходимость ввода данных, функция input(), понятие «значение функции», изменение типа данных при помощи функции int().

#### Практика

Отработка ввода команд с клавиатуры с использованием переменных.

### Тема 1.3. Строки

#### Теория

Интерпретатор. Строковый тип данных. Операции со строками: сложение строк, получение символа строки по индексу, срез, вхождение одной строки в другую, изменение регистра букв строки.

#### Практика

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).

## **Тема 1.4. Практикум по решению задач**

### Теория

Закрепление изученного на предыдущих уроках материала.

### Практика

Разработка программы на основе изученных материалов. Тренировка по оптимизацию качества кода программы.

## **Тема 2.1. Условный оператор**

### Теория

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

### Практика

Разработка программы с использованием условного оператора.

## **Тема 2.2. Вложенный условный оператор**

### Теория

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

### Практика

Разработка программ, содержащих вложенный условный оператор.

## **Тема 2.3. Циклы**

### Теория

Понятие «цикл». Цикл while. Переменная-счетчик. Цикл с накоплением результата вычислений в переменной. Цикл for. Функция range(). Отличие цикла for от цикла while. Цикл с вложенными условными операторами.

### Практика

Разработка программы, осуществляющей обработку данных с помощью цикла while. Разработка программы, осуществляющей обработку данных с помощью цикла for.

### **Тема 3.1. Функции**

#### Теория

Изучение понятий «функция», «определение функции», «параметры функции», «значение функции», «вызов функции». Синтаксис создания своих функций.

#### Практика

Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

### **Тема 3.2. Модули**

#### Теория

Изучение понятия «модуль», создание модуля, подключение модуля. Модуль random и time.

#### Практика

Написание программ на платформе, разработка проекта «Автоответчик».

### **Тема 3.3. Командный хакатон**

#### Теория

Составление чек-листов и Mind-Map. Самостоятельное тестирование и исправление ошибок в приложении.

#### Практика

Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

### **Тема 4.1. Списки и кортежи**

### Теория

Список. Кортеж. Итерируемый объект. Элементы списка. Индекс элемента. Создание пустого списка. Заполнение списка. Методы для работы со списками len(), append(), split(), join(), list(map(int, input().split())). Функция map(). Перебор элементов списка при помощи цикла.

### Практика

Разработка программы, осуществляющей обработку списка.

## **Тема 4.2. Словари и множества**

### Теория

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

### Практика

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

## **Тема 4.3. Вложенные структуры данных**

### Теория

Программирование вложенной структуры данных, применение подходящих методов к вложенной структуре, обращение к объекту во вложенной структуре, использование индексов (ключей).

### Практика

Решение задач повышенной сложности при помощи вложенной структуры.

## **Тема 4.4. Практикум по решению задач**

### Теория

Закрепление изученного на предыдущих уроках материала.

### Практика



Разработка программы на основе изученных материалов. Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

### **Тема 5.1. Классы**

#### Теория

Объекты и классы. Конструктор класса. Свойства и методы объектов.

Создание объектов.

#### Практика

Разработка программ, осуществляющих обработку классов.

### **Тема 5.2. Наследование**

#### Теория

Наследование как способ использовать в объектах готовые свойства и методы, добавляя к ним свои методы.

#### Практика

Разработка программ, осуществляющей наследование.

### **Тема 5.3. Игра Fast Clicker**

#### Практика

Проектная работа по созданию игры Fast Clicker.

### **Тема 5.4. Работа над проектами**

#### Теория

Технология работы над проектом. Итерационный подход. Составление чек-листов и Mind-Map. Самостоятельное тестирование проекта, взаимное ревью проектов учениками.

#### Практика

Создание проектов по выбору ученика.

Разработка программы на основе изученных материалов. Отработка ввода команд с клавиатуры, исправление ошибок в написанных программах и написание собственных программ на платформе.

### **Тема 5.1. Классы**

#### Теория

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

#### Практика

Разработка программ, осуществляющих обработку классов.

### **Тема 5.2. Наследование**

#### Теория

Наследование как способ использовать в объектах готовые свойства и методы, добавляя к ним свои методы.

#### Практика

Разработка программ, осуществляющей наследование.

### **Тема 5.3. Игра Fast Clicker**

#### Практика

Проектная работа по созданию игры Fast Clicker.

### **Тема 5.4. Работа над проектами**

#### Теория

Технология работы над проектом. Итерационный подход. Составление чек-листов и Mind-Map. Самостоятельное тестирование проекта, взаимное ревью проектов учениками.

#### Практика

Создание проектов по выбору ученика.

## 6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Для успешной реализации курса программирования Python Start необходимо соблюдение следующих условий:

Материально-технические условия реализации программы	<ul style="list-style-type: none"><li>- Учебный кабинет с местами для 12 учеников.</li><li>- Компьютеры для учеников (1 ученик — 1 компьютер) и компьютер для учителя.</li><li>- Проектор, подключенный к компьютеру для учителя, выводящий изображение на экран.</li><li>- Доступ в Интернет на каждом компьютере.</li><li>- Распечатанный список логинов и паролей учеников для доступа на платформу.</li><li>- Среда Visual Studio Code с интеграцией платформы, установленная на каждый компьютер.</li><li>- Доска или флипчарт.</li><li>- Листы бумаги А4.</li><li>- Ручки или карандаши (1 ученик — 1 ручка).</li></ul>
Учебно-методическое и информационное обеспечение программы	<ul style="list-style-type: none"><li>- Поурочные методические рекомендации к занятиям.</li><li>- Презентации к занятиям.</li><li>- Задания на платформе с автопроверкой.</li><li>- Задания на платформе со свободным решением.</li><li>- Бонусные задания на платформе.</li><li>- Задачи со свободным решением для реализации в среде программирования Visual Studio Code.</li></ul>

**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. 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.