

Комитет образования и науки администрации города Новокузнецка  
Муниципальное бюджетное учреждение  
дополнительного образования  
«Центр детского (юношеского) технического творчества «Меридиан»  
IT-CUBE.НОВОКУЗНЕЦК

РАССМОТРЕНА  
на заседании  
методического совета  
*Протокол № 04*  
*«26» июня 2020 г.*

ПРИНЯТА  
на заседании  
педагогического совета  
*Протокол № 05*  
*«26» июня 2020 г.*



**Дополнительная общеобразовательная общеразвивающая программа**

**«Программирование на Python»**

технической направленности

Возраст учащихся: 14-16 лет

Срок реализации: 1 год (144 часа)

Разработчик: Смагин К.Н.,  
педагог дополнительного образования

2020 год

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «**Программирование на Python**» относится к программам **технической направленности базового уровня**.

Нормативные документы, на основании которых разработана программа:

- Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Распоряжение Правительства РФ от 4 сентября 2014 г. № 1726-р «Концепция развития дополнительного образования детей»;
- Приказ Министерства просвещения РФ от 9 ноября 2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Распоряжение Министерства просвещения РФ от 1 марта 2019 г. № Р-24 «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «ИТ-Куб» в рамках федерального проекта «Цифровая образовательная среда» национального проекта «Образование»;
- Постановление Главного государственного санитарного врача РФ от 04.07.2014 № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы ОО ДОД»;
- Приказ Департамента образования и науки Кемеровской области от 05.05.2019 г. № 740 «Об утверждении Правил персонифицированного финансирования дополнительного образования детей»;
- Устав МБУ ДО «Центр детского (юношеского) технического творчества «Меридиан».

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

**Актуальность программы.** В рамках изучения программы обучающиеся постоянно будут сталкиваться с необходимостью самостоятельной работы над заданиями: школьники учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых для решения задачи необходимо найти какую-то информацию в сети Интернет; может потребоваться устранение ошибки, которую не так просто обнаружить; условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю). Все эти знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят школьников к самостоятельной проектно-исследовательской деятельности с применением современных технологий.

**Педагогическая целесообразность** программы заключается в том, что она предполагает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном

уровне; имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика; охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний; ориентирована на существующий парк вычислительной техники и дополнительные ограничения; допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального), а также предусматривает возможность индивидуальной работы с учащимися.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения. Осваивая данную программу, обучающиеся будут овладевать навыками востребованных уже в ближайшие десятилетия специальностей, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, рассматриваемые в программе.

**Цель программы:** подготовка школьников к использованию методов программирования на языке Python в учебной и последующей профессиональной деятельности, расширение кругозора обучающихся в различных предметных областях.

**Задачи:**

- познакомить с конструкцией языка программирования Python и работой в интегрированной среде разработки на языке Python;
- формировать навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- развивать навыки алгоритмического и логического мышления, грамотной разработки программ, поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python;
- развивать инженерное мышление, изобретательность, умение самостоятельно решать поставленную задачу;
- содействовать профессиональному самоопределению обучающихся;
- воспитывать упорство в достижении поставленной цели и результата.
- воспитывать навыки самоорганизации; самостоятельной и командной работы.

Обучение по данной программе основано на следующих **принципах:** научности, сознательности, доступности, наглядности, последовательности, связи теории с практикой, вариативности.

**Отличительная особенность программы.** Программа организована по принципу дифференциации по уровням сложности и состоит из трех блоков, имеющих логическую завершенность: «Введение в программирование» - позволяет обеспечить начальную подготовку детей в области программирования и формирует положительную мотивацию к языкам программирования; «Базовые конструкции языка Python» и «Решение прикладных задач в Python» - предполагает освоение специализированных знаний в языке программирования Python.

Программа «Программирование на Python» рассчитана на 144 часа, реализуется в течение 1 года на базе ИТ-Куба в учебном кабинете с необходимым оборудованием, техническим и ресурсным обеспечением в соответствии с перечнем, указанным в методических рекомендациях по созданию и функционированию центров цифрового образования «ИТ-Куб» в рамках федерального проекта «Цифровая образовательная среда» национального проекта «Образование». Занятия проводятся по 2 часа 2 раза в неделю.

Программа предназначена для обучающихся 14-16 лет, которые имеют минимальный необходимый уровень входных компетенций: уверенный пользователь ПК, проявляют интерес к программированию. Количество детей в группе от 7 до 12 человек. Реализация программы допускает разновозрастной состав учащихся, что способствует социальному развитию детей, формированию умения работать в разновозрастном коллективе.

**Формы и методы работы** фронтальные, групповые, индивидуальные: беседа, объяснение, практические работы, инструктаж, самостоятельная работа (индивидуально и в малых группах), воркшопы, участие в профильных мероприятиях и соревнованиях, демонстрация наглядного материала, мозговой штурм, частично-поисковый (эвристический) метод, исследовательский метод, метод проектов; метод проблемного изложения; устный опрос, публичное выступление и др.

Основной тип занятий – комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Каждая тема начинается с постановки задачи – характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью педагог проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый учащийся получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

## **Планируемые результаты**

### **Предметные и предпрофессиональные результаты (hard компетенции)**

В результате освоения программы учащиеся

#### **будут знать:**

- технику безопасности при нахождении в IT-Кубе, работе со специальным оборудованием при выполнении практико-ориентированных заданий;
- правила безопасной работы на компьютере;
- основные предметные понятия («информация», «алгоритм», «исполнитель», «модель») и их свойства;
- основные алгоритмические структуры – линейную, условную и циклическую;
- этапы разработки проектов; правила презентации и продвижения проектного продукта;

#### **будут уметь:**

- организовывать рабочее место;
- соблюдать технику безопасности, технологически правильно обращаться с оборудованием IT-Куба и инструментами при выполнении практико-ориентированных работ, следовать требованиям гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий;
- определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- составлять и записывать алгоритм для конкретного исполнителя;

- выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- разрабатывать программу в выбранной среде программирования, включая тестирование и отладку программ;
- использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- искать информацию с применением правил поиска в компьютерных сетях, не компьютерных источниках информации (справочниках и словарях, каталогах, библиотеках) при выполнении заданий и проектов по различным темам;
- следовать требованиям техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий.

### **Личностные и метапредметные результаты (soft компетенции)**

#### **Личностные**

- деятельности и принципов тайм-менеджмента;
- умение использовать приобретенные знания и умения в практической деятельности и повседневной жизни для:
  - создания простейших моделей объектов и процессов в виде изображений и чертежей,
  - создания информационных объектов, в том числе для оформления результатов учебной работы;
  - организации индивидуального информационного пространства, создания личных коллекций информационных объектов;
  - передачи информации по телекоммуникационным каналам в учебной и личной переписке, использования информационных ресурсов общества с соблюдением соответствующих правовых и этических норм;
- знание техники ведения проектной работы, владение основными универсальными умениями информационного характера (постановка и формулирование проблемы, поиск и выделение необходимой информации, выбор наиболее оптимальных способов решения задач в зависимости от конкретных условий);
- постановка цели собственного развития, соотносить собственные возможности и поставленные задачи, определять способы действий в рамках предложенных условий, осуществлять контроль своей деятельности, объективно оценивать результаты своей работы, соотносить свои действия с планируемыми результатами;
- навыки самопрезентации.

#### **Метапредметные**

##### **регулятивные**

- умение осуществлять целеполагание, планирование, корректировку плана, прогнозирование, контроль, коррекцию, оценку деятельности;
- искать информацию с применением правил поиска в компьютерных сетях, некомпьютерных источниках информации (справочниках и словарях, каталогах, библиотеках) при выполнении заданий и проектов по различным темам;

##### **познавательные**

- умение поставить учебную задачу, выбрать способы и найти информацию для её решения;
- умение работать с информацией, структурировать полученные знания;
- умение анализировать и синтезировать новые знания, устанавливать причинно-следственные связи, доказывать свои рассуждения;

- умение сформулировать проблему и найти способы её решения;
- коммуникативные
- командные компетенции и умение работать в команде;
  - умение слушать и слышать собеседника, аргументировать свою точку зрения;
  - умение осуществлять инициативное сотрудничество в поиске и сборе информации;
  - навыки публичного выступления и презентации результатов.

### **Формы контроля и подведения итогов реализации программы**

На занятиях используются: входной и текущий контроль, промежуточная и итоговая аттестация.

Входной контроль осуществляется через наблюдение за деятельностью учащихся, предполагает собеседование с учащимися, в ходе которого определяется наличие у них минимального необходимого уровня входных компетенций: уверенный пользователь ПК, проявляет интерес к прикладному программированию, конструированию, мехатронике, информационным технологиям в целом.

Текущий контроль осуществляется посредством наблюдения за деятельностью учащихся на каждом занятии и фиксации их умений во время работы над практическими заданиями/работами по разделам и мини проектами. Отмечается активность участия учащихся в мероприятиях, степень самостоятельности при работе над практическими заданиями, самостоятельный поиск и разработка интересных тем для доклада (или мини-проекта).

Промежуточная и итоговая аттестация предполагает написание программы для решения одной из задач, контрольные работы и задания по изученным темам для определения уровня знаний учеников, разработку и реализацию проектов, представление и защиту индивидуальных и групповых проектов, публичное выступление с демонстрацией результатов работы, участие в профильных конкурсах и мероприятиях.

### **УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

№	Тема	Количество часов			Формы контроля/ аттестации
		всего	теория	практика	
1.	<b>Вводное занятие</b>	<b>2</b>	<b>1</b>	<b>1</b>	Собеседование
2.	<b>Введение в программирование</b>	<b>26</b>	<b>8</b>	<b>18</b>	Беседа, выполнение мини-проекта, практические работы
	2.1. Понятия кода, интерпретатора, программы	8	2	6	
	2.2. Интегрированные среды, исполнение кода и отладка	10	2	8	
	2.3. Переменные, основные операторы	6	2	4	
	2.4. Базовые типы данных, ветвления	2	2		
3.	<b>Базовые конструкции в Python</b>	<b>40</b>	<b>6</b>	<b>34</b>	Практические задания, беседа, выполнение мини-проекта, контрольная работа
	3.1. Циклы, срезы, списочные выражения	12	2	10	
	3.2. Методы списков и строк. Функции	8	2	6	
	3.3. Решение задач по пройденным темам. Практическая работа	10	2	8	
	3.4. Контрольная работа по темам	10		10	
4.	<b>Решение прикладных задач в Python.</b>	<b>76</b>	<b>20</b>	<b>56</b>	Практические задания, беседа, выполнение мини-проекта,
	4.1. Функции (углублённое рассмотрение), другие структуры	40	8	32	

данных, библиотеки Python, введение в ООП				контрольная работа
4.2. Решение задач на пройденные темы	20	8	12	
4.3. Проверочные работы и итоговая контрольная работа	16	4	12	
Всего:	144	35	85	

## СОДЕРЖАНИЕ ПРОГРАММЫ

### Раздел 1. Вводное занятие - 2 часа.

Общая информация об IT-Кубе, актуальность направления. Представление программы, ожиданий участников, правил работы. Профильные мероприятия, конкурсы, соревнования. Вводный инструктаж по технике безопасности. Правила работы в объединении и организации рабочего места. Языки программирования. Знакомство участников (индивидуальная презентация, знакомство в малых группах, игры и др.).

**Практическая работа.** Экскурсия по IT-Кубу. Собеседование. Установка и настройка ПО, необходимое для разработки на языке.

### Раздел 2. Введение в программирование – 26 часов.

#### Тема 2.1. Понятия кода, интерпретатора, программы – 8 часов.

Понятия кода, интерпретатора, программы. Простейшие программы с выводом на экран. Знакомство с виртуальной средой взаимодействия – регистрация, организация личного кабинета, поиск и выкладывание материалов; знакомство с системой автоматизированной проверки задач и системой Яндекс.

**Практическая работа.** Решение задач.

#### Тема 2.2. Интегрированные среды, исполнение кода и отладка – 10 часов.

Интегрированные среды, исполнение кода. Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки.

**Практическая работа.** Решение задач.

#### Тема 2.3. Переменные, основные операторы – 6 часов.

Условный оператор. Переменные и арифметика. Погружение в условия. Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

**Практическая работа.** Решение задач.

#### Тема 2.4. Базовые типы данных, ветвления – 2 часа.

Знакомство со списками, строками, множествами и кортежами в Python.

**Практическая работа.** Разработка алгоритмов и программ, определение работоспособности разработанной программы.

### Раздел 3. Базовые конструкции в Python – 40 часов.



### **Тема 3.1. Циклы, срезы, списочные выражения – 12 часов.**

Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python. Знакомство со срезами и диапазонами. Равенство и совпадение объектов. Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

*Практическая работа.* Решение задач по теме «Срезы и диапазоны».

### **Тема 3.2. Методы списков и строк. Функции – 8 часов.**

Списочные выражения. Методы split и join. Другие методы списков и строк. Знакомство с функциями. Области видимости переменных. Работа со списками, строками, множествами и кортежами в Python. Понятие итератора, подпрограммы, процедуры, функции. Функции в Python.

Практика: решение задач.

### **Тема 3.3. Решение задач по пройденным темам – 10 часов.**

Повторение пройденных тем.

*Практическая работа.* Решение задач и выполнение практических и творческих заданий по пройденным темам.

### **Тема 3.4. Контрольная работа по темам – 10 часов.**

*Практическая работа.* Решение задач контрольной работы на методы списков и строк, по темам «Методы split и join», «Функции», «Области видимости переменных». Анализ результатов.

## **Раздел 4. Решение прикладных задач в Python – 76 часов.**

### **Тема 4.1. Функции (углублённое рассмотрение), другие структуры данных, библиотеки Python, введение в ООП – 40 часов.**

Функции, функциональная парадигма программирования. Понятие ассоциативного массива. Словари и множества, модульный принцип компоновки программы. Подключение и использование модулей стандартной библиотеки Python, дополнительные библиотеки. Понятие репозитория различных пакетов Python. Основы ООП.

*Практическая работа.* Работа с документацией в стандартной библиотеке, с внешними библиотеками Python и утилитой pip.

### **Тема 4.2. Решение задач на пройденные темы – 20 часов.**

Прикладные задачи.

*Практическая работа.* Решение задач по темам «Функции и функциональная парадигма программирования», «Словари и множества», «Стандартная библиотека Python», «Дополнительные библиотеки Python», «Введение в ООП». Самостоятельная работа: работа с конспектом, который описывает способы решения задач, разработка проекта по индивидуальному заданию, составление отчета о выполнении индивидуальной или совместной работы.

Тематика самостоятельных работ:

- задачи, в которых по заданному алгоритму необходимо написать программу;
- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;
- задачи, для решения которых необходимо найти некоторую информацию в Интернете.

Выполненные самостоятельные работы загружаются в автоматизированную систему проверки задач на Яндекс.

### **Тема 4.3. Проверочные работы и итоговая контрольная работа – 16 часов.**

Повторение изученного материала, подготовка к контрольной работе. Разбор контрольной работы.

*Практическая работа.* Выполнение проверочных работ. Итоговая контрольная работа по изученным темам. Разработка проекта по индивидуальному заданию, составление отчёта о выполнении индивидуальной или совместной работы.

## **МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

Раздел программы	Формы занятий	Приёмы и методы организации образовательно-	Дидактический материал	Техническое оснащение занятий	Формы подведения итогов
------------------	---------------	---	------------------------	-------------------------------	-------------------------

		го процесса			
Вводное занятие	Лекция, презентация, игра, инструктаж	Словесно-наглядный, проблемное изложение, поиск ответов на поставленные вопросы	Презентация, инструкции, подборка профильных мероприятий	Оборудование ИТ-Куба	Собеседование
Введение в программирование	Лекция, групповая, индивидуальная, практическая работа, проекты	Словесно-наглядный, поисковый, практический, проблемный	Подборка презентационных материалов и прикладных задач	Оборудование ИТ-Куба	Беседа, выполнение мини-проекта, практические работы
Базовые конструкции в Python	Лекция, групповая, индивидуальная, практическая работа, решение задач, проекты	Словесно-наглядный, поисковый, практический, проблемный	Подборка презентационных материалов и прикладных задач	Оборудование ИТ-Куба	Практические задания, беседа, выполнение мини-проекта, контрольная работа
Решение прикладных задач в Python.	Лекция, групповая, индивидуальная, практическая работа, решение задач, проекты	Словесно-наглядный, поисковый, практический, проблемный	Подборка презентационных материалов и прикладных задач	Оборудование ИТ-Куба	Практические задания, беседа, выполнение мини-проекта, контрольная работа

## МАТЕРИАЛЬНО – ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Условия реализации программы: учебный кабинет, оснащенный оборудованием (стандарт).

Перечень необходимого оборудования и расходных материалов (количество единиц оборудования и материалов указано из расчета на 12 человек):

- компьютеры и ноутбуки, на которых установлено соответствующее программное обеспечение: на каждого обучающегося и преподавателя - 12 шт. или 1 шт. на малую группу (должны быть подключены к единой Wi-Fi сети с доступом в интернет);
- презентационное оборудование – 2 шт.;
- интерактивная панель – 1 шт.;
- источники бесперебойного питания - PowerMan Back PRO 800 Plus – 12 шт.;
- наушники полноразмерные - Sennheiser HD 206 – 12 шт.;
- акустическая система - Bluetooth SVEN SPS-721 – 1 шт.;

Программное обеспечение:

- операционная система – любая, желательно Windows;
- поддерживаемые браузеры для работы LMS:
- Yandex Browser 17+
- Chrome 60+
- Chrome Mobile 60+
- Firefox 52+
- Opera 50+
- Safari 11+
- Mobile Safari 11+
- Edge 16+
- Python 3.6.0 или выше;
- нужно прописать путь в PATH для запуска Python из консоли (обычно это можно сделать с помощью установщика);
- нужно проверить, что утилита pip корректно работает и есть возможность устанавливать дополнения без прав администратора;
- среда Wing IDE 101 версии 6 или выше для Python;

- среда PyCharm Community Edition;
- пакет PyQt4 (на Qt5);
- пакет библиотек со SciPy: numpy, scipy, matplotlib, ipython + ipythonnotebook, sympy, pandas;
- рекомендуется установить ПО Anaconda, это поможет решить все вопросы с пакетами.

## **ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ**

### **СПИСОК ЛИТЕРАТУРЫ ДЛЯ ПЕДАГОГА**

1. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
3. Лутц, М. Изучаем Python / М. Лутц. - СПб.: Символ-Плюс, 2011.
4. Окулов С. М. Основы программирования / С. М. Окулов. - М.: Бином. Лаборатория знаний, 2012.
5. Поляков, К. Ю. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях / К. Ю. Поляков, Е. А. Еремин. – М.: БИНОМ. Лаборатория знаний, 2014.

### **ИНТЕРНЕТ – РЕСУРСЫ**

1. Материалы и презентации к урокам в LMS Яндекс.Лицея.
2. Сайт [pythonworld.ru](http://pythonworld.ru) – «Python 3 для начинающих».
3. Сайт [pythontutor.ru](http://pythontutor.ru) – «Питонтьютор».
4. <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl> – Лекции А.В. Умнова, прочитанные в Школе анализа данных Яндекса.

**Лист оценки работы обучающихся  
в процессе разработки**

№ группы: \_\_\_\_\_

Дата: \_\_\_\_\_

№ п/п	ФИ уч-ся	Сложность языка разработки (по шкале от 0 до 5 баллов)	Соответствие написания программного кода поставленной задаче (по шкале от 0 до 5 баллов)	Презентация модели по плану. Степень владения специальными терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригинальности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруднений (шт. за одно занятие)
1.						
2.						

**Лист оценки работы обучающихся  
в процессе настройки коммутационного и компьютерного оборудования**

№ п/п	ФИ уч-ся	Соответствие построенной конструкции заданной модели (по шкале от 0 до 5 баллов)	Соответствие написанной программы заданным целям (по шкале от 0 до 5 баллов)	Степень владения специальными терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригинальности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруднений (шт. за одно занятие)
1.						
2.						