

**Муниципальная бюджетная общеобразовательная учреждение
«Лицей-интернат (школа для одарённых детей) г. Буинска Республики
Татарстан»**

«Рассмотрено»
Руководитель МО классных
руководителей

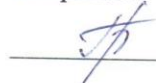
 /Л.М.Бакунина/

Протокол № 1 от
« ____ » августа 2020 г.

«СОГЛАСОВАНО»

Заместитель руководителя по

ВР МБОУ «Лицей интернат
(школа для одаренных детей)
г.Буинска Республики
Татарстан»

 /Г.А. Галиева/

« 31 » августа 2022 г.

«УТВЕРЖДАЮ»

Директор МБОУ «Лицей
интернат (школа для одаренных
детей) г.Буинска Республики
Татарстан»



/И.И.Абзалов/

Приказ № 155 от

« 31 » августа 2022 г.

**РАБОЧАЯ ПРОГРАММА
Дополнительного образования
технической направленности
Программирование на языке Python**

Составитель: Шамгунова Гульнара Наилевна
учитель информатики

**Возраст учащихся: 12-17 лет.
Срок реализации: 1 год (156 часов).**

Буинск, 2022 год

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ

Наименование программы	Рабочая программа дополнительного образования «Основы программирования на языке Python» с использованием оборудования центра естественно-научной и технологической направленностей «Точка роста»
Адресность программы	Программа адресована обучающимся 6-9 классы МБОУ «Лицей-интернат г.Буинска республики Татарстан»
Разработчик программы	Шамгунова Гульнара Наилевна, учитель информатики

Пояснительная записка

На базе центра «Точка роста» обеспечивается реализация образовательных программ технологической направленности, разработанных в соответствии с требованиями законодательства в сфере образования и с учётом рекомендаций Федерального оператора учебного предметов «Информатика» и «Технология».

Программа «Основы языка программирования Python» составлена на основе курса Д.П. Кириенко «Основы языка программирования Python», М.: Бином, 2014 г., Примерной программы внеурочной деятельности начального и основного образования, Стандартов второго поколения в форме кружка в 6-9 классах.

Программа «Основы программирования на языке «Python» направлена на подготовку творческой, технически грамотной, гармонично развитой личности, обладающей логическим мышлением, способной анализировать и решать задачи в команде в области информационных технологий, решать ситуационные кейсовые задания, основанные на групповых проектах. Занятия по данному курсу рассчитаны на общенаучную подготовку обучающихся, развитие их мышления, логики, математических способностей, исследовательских навыков. Курс направлен на изучение основ программирования на языке Python. В рамках курса обучающиеся смогут познакомиться с физическими, техническими и математическими понятиями. Приобретённые знания будут применимы в творческих проектах.

Актуальность курса состоит в том, что он направлен на:

- ✓ формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики за счет развития представлений об информации как важнейшем стратегическом ресурсе развития личности, государства, общества; понимания роли информационных процессов в современном мире;
- ✓ совершенствование общеучебных и общекультурных навыков работы с информацией в процессе систематизации и обобщения имеющихся и получения новых знаний, умений и способов деятельности в области информатики и ИКТ; развитию навыков самостоятельной учебной деятельности школьников (учебного проектирования, моделирования, исследовательской деятельности и т.д.);
- ✓ воспитание ответственного и избирательного отношения к информации с учётом правовых и этических аспектов ее распространения, воспитанию стремления к продолжению образования и созидательной деятельности с применением средств ИКТ.

Использование оборудования «Точка роста» при реализации данной образовательной программы позволяет создать условия:

- ✓ для расширения содержания школьного образования по информатике и технологии;
- ✓ для повышения познавательной активности обучающихся в области программирования;
- ✓ для развития личности ребёнка в процессе обучения информатики и технологии, его способностей, формирования и удовлетворения социально значимых интересов и потребностей;
- ✓ для работы с одарёнными школьниками, организации их развития в различных областях образовательной, творческой деятельности.

Общая характеристика программы

Программа по предмету «Основы языка программирования Python» предназначена для изучения всех основных разделов курса программирования на базовом уровне.

В рамках предлагаемого курса изучение основ программирования на языке Python — это не столько средство подготовки к будущей профессиональной деятельности, а сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Исключительно велика роль программирования для формирования мышления школьников, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления учащихся. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции — одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных школьников.

Цель изучения курса:

формирование интереса обучающихся к изучению профессий, связанных с основами программирования через освоение языка Python.

Задачи:

обучающие:

- ✓ освоение основных алгоритмических конструкций;

- ✓ обучение основам алгоритмизации и программирования;
- ✓ освоение первоначальных навыков программирования на языке программирования высокого уровня Python;
- ✓ приобщение к проектно-творческой деятельности;

воспитывающие:

- ✓ воспитание интереса к информационной и коммуникационной сфере человеческой деятельности,
- ✓ воспитание потребности соблюдать этические и правовые нормы работы с информацией;
- ✓ воспитание бережного отношения к техническим устройствам;

развивающая:

- ✓ развитие творческого воображения, алгоритмического мышления учащихся;
- ✓ развитие навыков планирования проекта, умения работать в группе;
- ✓ развитие навыков ориентации в информационных потоках окружающего мира и применения точной и понятной инструкции для решения учебных задач и в повседневной жизни.

Планируемый результат реализации программы

Основным результатом обучения является формирование вектора развития обучающихся с упором на формирование у них алгоритмического мышления.

Личностные результаты:

- ✓ сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
- ✓ осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов;
- ✓ сформированность представлений о мире профессий, связанных с программированием, и требованиях, предъявляемых различными востребованными профессиями, такими как программист, системный администратор;
- ✓ навыки сотрудничества в образовательной, учебно-исследовательской, проектной и других видах деятельности;
- ✓ навыки взаимо- и самооценки, навыки рефлексии.

Метапредметные результаты:

- ✓ владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;

- ✓ способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания.

Предметные результаты:

- ✓ навыки алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- ✓ владение стандартными приёмами написания программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ на языке Python;
- ✓ знание особенностей структуры программы, представленной на языке Python,
- ✓ представление о модулях, входящих в состав среды Python,
- ✓ возможности и ограничения использования готовых модулей,
- ✓ представление о величине, ее характеристиках,
- ✓ знание что такое операция, операнд и их характеристики,
- ✓ знание принципиальные отличия величин, структурированных и не структурированных,
- ✓ представление о таких структурах данных, как число, текст, кортеж, список, словарь,
- ✓ представление о составе арифметического выражения;
- ✓ знание математических функций, входящих в Python, представление о логических выражениях и входящих в них операндах, операциях и функциях,
- ✓ умение записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить,
- ✓ знание основных операторов языка Python, их синтаксис,
- ✓ представление о процессе исполнения каждого из операторов,
- ✓ умение разрабатывать программы обработки числовой и символьной информации,
- ✓ умение разрабатывать программы (линейные, разветвляющиеся и с циклами),
- ✓ представление о значении полноценных процедур и функций для структурно-ориентированного языка высокого уровня,
- ✓ правила описания функций в Python и построение вызова,
- ✓ принципиальные отличия между формальными, локальными и глобальными переменными.

Сроки реализации программы

Программа рассчитана на 156 учебных занятий. Продолжительность занятия – 2 часа.

Форма и режим занятий

Основными видами учебной деятельности учащихся является компьютерный практикум и компьютерный эксперимент по предложенным учебным материалам. Основная форма обучения: практические работы на компьютере. Режим занятий – 4 часа в неделю.

Формы подведения итогов реализации программы

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются образовательные продукты учащихся (созданные блок-схемы, программы), а также личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество образовательной продукции оценивается по следующим параметрам:

- ✓ алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- ✓ программа должна выполнять поставленные задачи;
- ✓ по степени «читаемости кода» (должны быть соблюдены отступы, обязательно наличие комментариев к коду программы и т. д.).

Проверка достигаемых учащимися результатов производится в следующих формах:

- ✓ текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- ✓ текущая диагностика и оценка учителем деятельности школьников;
- ✓ итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальной образовательной программой освоения курса.

Итоговый контроль проводится в конце всего курса в форме тестирования.

Содержание программы Учебный план

№ п/п	Наименование тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	

Инструктаж по технике безопасности в компьютерном классе		2	2		Опрос
1.	Раздел 1. Знакомство с языком Python	6	2	4	Тестирование, решение практических задач
1.1	Занятие 1. Общие сведения о языке Практическая работа: Установка программы Python	2	1	1	
1.2.	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	2	1	1	
1.3	Тест № 1. Знакомство с языком Python	2		2	
2.	Раздел 2. Переменные и выражения	11	4	7	Тестирование, решение практических задач
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	2	1	1	
2.2	Практическая работа: Переменные	2		2	
2.3	Занятие 4. Выражения Практическая работа: Выражения	2	1	1	
2.4	Занятие 5. Ввод и вывод	1	1		
2.5	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	2	1	1	
2.6	Тест № 2. Выражения и операции.	2		2	
3	Раздел 3. Условные предложения	28	6	22	Тестирование, Решение практических задач
3.1	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	2	1	1	
3.2	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	4	1	3	

3.3	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление	4	1	3	
3.4	Занятие 10. Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	2	1	1	
3.5	Самостоятельная работа № 1 по теме «Условные операторы».	10	2	8	
3.6	Занятие 11. Зачетная работа № 1. "Составление программ с ветвлением".	4		4	
3.7	Тест № 3. "Условные операторы".	2		2	
4	Раздел 4. Циклы	46	9	37	Тестирование, решение практических задач, творческая работа
4.1.	Занятие 12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	4	1	3	
4.2.	Занятие 13. Оператор цикла for Практическая работа Решение задачи с циклом for.	4	1	3	
4.3.	Занятие 14. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	4	1	3	
4.4.	Занятие 15. Случайные числа Практическая работа: Случайные числа	4	1	3	
4.5.	Занятие 16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	6	1	5	
4.6	Самостоятельная работа № 2 "Составление программ с циклом"	10	2	8	
4.7	Тест № 4. Циклы	2		2	
4.8	Занятие 17. Творческая работа № 1. "Циклы"	12	2	10	
5	Раздел 5. Функции	20	6	14	Тестирование,

5.1.	Занятие 18. Создание функций Практическая работа Создание функций	2	1	1	решение практических задач
5.2.	Занятие 19. Локальные переменные Практическая работа Локальные переменные	2	1	1	
5.3.	Занятие 20. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2	1	1	
5.4	Самостоятельная работа № 3 по теме "Функции"	10	2	8	
5.5	Занятие 21. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	2	1	1	
5.6	Тест № 5. Функции	2		2	
6	Раздел 6. Строки - последовательности символов	10	3	7	Решение практических задач
6.1.	Занятие 22. Строки Практическая работа: Строки	2	1	1	
6.2.	Занятие 23. Срезы строк	2	1	1	
6.3	Занятие 24. Примеры решения задач со строками Практическая работа: Решение задач со строками.	6	1	5	
7	Раздел 7. Сложные типы данных	17	7	10	Тестирование, Решение практических задач
7.1.	Занятие 25. Списки Практическая работа: Списки	2	1	1	
7.2	Занятие 26. Срезы списков	1	1		
7.3	Занятие 27. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	6	1	5	
7.4	Занятие 28. Матрицы	2	1	1	
7.5	Тест № 7. Списки	2		2	
7.6	Занятие 29. Кортежи	1	1		

7.7	Занятие 30. Введение в словари	1	1		
7.8	Занятие 31. Множества в языке Python	2	1	1	
8	Раздел 8. Стиль программирования и отладка программ	17	7	10	Решение практических задач, презентация проекта, рефлексия.
8.1	Занятие 32. Стиль программирования	2	1	1	
8.2	Занятие 33. Отладка программ	2	1	1	
8.3	Занятие 34. Зачет по курсу «Программирование на языке Python»	10	2	8	
8.3	Занятие 35. Что дальше?	2	2		
	ВСЕГО	156	45	111	

Содержание учебного плана

Вводное занятие (2 часа).

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python (6 часов)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест № 1. Знакомство с языком Python

Учащиеся должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения (11 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2.

Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами Тест № 2.

Выражения и операции.

Учащиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения (28 часов)

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы". Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

Учащиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы (46 часов)

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

Учащиеся должны знать / понимать:

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции (20 часов)

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция `lambda`. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции" Тест № 5. Функции

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки - последовательности символов (10 часов)

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор `in`. Модуль `string`. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать / понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Раздел 7. Сложные типы данных (17 часов)

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест № 7. Списки

Учащиеся должны знать / понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

Раздел 8. Стиль программирования и отладка программ (15 часов)

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать / понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

Планируемые результаты курса

В рамках курса «Программирование на языке Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

Проверка достигаемых учащимися результатов производится в следующих формах:

- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка педагогом деятельности учащихся;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;
- Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1. Календарный учебный график

Работа по программе осуществляется с 01 сентября по 30 мая.

2.2. Условия реализации программы

Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- специальная и научно-популярная литература для педагога и учащихся;

- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство не менее 128 Гб;
- Монитор диагональю не менее 19”;

Программное обеспечение:

- Операционная система Windows 10 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JupyterLab PyCharm;
- Foxit Reader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Растровый графический редактор;
- Любой браузер для интернет серфинга.

Список литературы

Нормативно-правовые акты и документы

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ
2. Приказ Министерства образования и науки РФ от 29.08.2013 №1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Письмо Министерства образования Российской Федерации от 18 июня 2003 №28-02-484/16 «Требования к содержанию и оформлению образовательных программ дополнительного образования детей»,
4. Постановление Главного государственного санитарного врача Российской Федерации от 4.07.2014 г. «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Стратегия инновационного развития Российской Федерации на период до 2020 года. Распоряжение правительства Российской Федерации от 08.12.2011 № 2227-р.
6. Концепция развития дополнительного образования детей. Распоряжение Правительства Российской Федерации от 24 апреля 2015 года № 729-р.

7. Стратегия развития воспитания в Российской Федерации на период до 2025 года». Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

Литература для родителей и учащихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
 2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
 3. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
 4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.
- Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net , свободный

В этом документе пронумеровано,
проиндексировано и скреплено
печатью

И.И. Абзалов
Директор

