

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ  
«ШКОЛА С УГЛУБЛЕННЫМ ИЗУЧЕНИЕМ МАТЕМАТИКИ №1360»

Принята на заседании  
методического (педагогического)  
совета  
Протокол № 1  
от «26» 08 2016 года



«Утверждаю»  
Директор ГБОУ Школа №1360  
Синченко Л.А.

Приказ № 1/74-09  
от «01» 09 2016 года

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**  
**«Олимпиадная информатика»**  
Направленность: техническая  
Уровень программы: ознакомительный  
Возраст учащихся: 14-18 лет  
Срок реализации: 1 год (105 часов)

**Автор-составитель:**  
Глинка Надежда Владимировна,  
педагог дополнительного образования

Москва 2016

## Пояснительная записка

В условиях бурного развития IT технологий, благосостояние страны и ее жителей все более зависит от умения работать с информацией и умения эту информацию защищать. Можно долго спорить о том, что уже все давно запрограммировали и современная разработка прикладного программного обеспечения, зачастую не требует знания языков программирования. Что мы видим на самом деле. Развитые страны в своих образовательных программах начинают делать упор на освоение алгоритмизации и программирования у подростков 12-15 лет. Принимают специальные законы. И все больше специалистов понимает, что изучение основ программирования связано с целым рядом умений и навыков (организация деятельности, планирование ее и т.д.), которые по праву носят общеинтеллектуальный характер и формирование которых - одна из приоритетных задач современной школы.

Поэтому повышается значимость программ дополнительного образования, связанных с программированием и созданием новых алгоритмов.

Очень велика роль изучения программирования для развития мышления школьников, формирования многих приемов умственной деятельности. Поэтому не использовать действительно большие возможности программирования, формирования многих общеучебных, общеинтеллектуальных умений и навыков было бы неправильно.

Изучая программирование, учащиеся приобщаются к алгоритмической культуре, познают азы профессии программиста. Олимпиады по Информатике направлены на развитие у школьников информационной культуры и грамотности.

Разработанная программа предназначена для учащихся 14-18 лет, срок выполнения – 1 год, 105 академических часов, по 3 часа в неделю.

### Цели и задачи

**Целями** курса дополнительного образования «Олимпиадная информатика» являются:

1. формирование актуальных компетентностей в области IT;
2. осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов;
3. развитие алгоритмического мышления учащихся;
4. отношение к профессиональной деятельности в IT-сфере как возможности участия в решении личных, общественных, государственных, общенациональных и общечеловеческих проблем;

**Задачами** курса «Олимпиадная информатика» являются:

1. освоение всевозможных методов решения задач, реализуемых на языке Python;
2. изучить специальные приемы программирования и решение практических задач повышенной сложности, используя особенности языка программирования Python;

5. углубление знаний, умений и навыков решения задач по программированию и алгоритмизации;
6. формирование навыков грамотной разработки программ;
3. тренинг по решению задач олимпиадного характера развить алгоритмическое мышление учащихся.

Для подведения итогов и промежуточного контроля реализации программы предусмотрены различные виды аттестации: в течение года – промежуточные зачетные работы, контроль в тестирующей системе, олимпиады.

## Содержание программы

### Тематический план

**Тема 1.** Особенности языка программирования Python. Компиляторы и интерпретаторы. Среды программирования для языка Python.

**Тема 2.** Как писать программы на языке программирования Python.

**Тема 3.** Данные, их типы. Арифметические операции, целочисленная арифметика. Изменение типа данных. Переменные. Логические операции над данными.

**Тема 4. Условный оператор.** Инструкция if. Более сложная форма условия: if – else. Множественное ветвление (elif).

**Тема 5. Цикл for.** Цикл **for** – это инструкция, выполняющая одну и ту же последовательность действий. Синтаксис. Практика использования.

**Тема 6. Цикл с условием while.** Цикл **while** – это инструкция, выполняющая одну и ту же последовательность действий, пока выполняется определенное условие. Синтаксис. Практика использования.

**Тема 6.** Действительные (вещественные) числа. Библиотека Math. Практические задачи.

**Тема 7. Строки.** Объявление строкового типа данных. Особенности работы со строковыми типами данных на языке Python. Срезы. Методы. Практические задачи.

**Тема 8. Символы и строки.** Таблица кодов ASCII. Функции ord и chr. Практические задачи.

**Тема 9. Функции.** Это изолированный блок кода, обращение к которому в течение выполнения программы может быть многократным. Для чего пишут функции. Локальные и глобальные переменные. Стандартные функции. Синтаксис. Практические задачи.

**Тема 10. Рекурсия.** Функция может вызывать и саму себя. Синтаксис. Простые примеры рекурсивных определений и рекурсивных подпрограмм. Практические задачи. Решение сложных задач с использованием рекурсии.

**Тема 11. Последовательности данных или одномерные списки.** Понятие списка. Способы объявления списков. Ввод-вывод элементов списка.

**Тема 12. Задачи обработки списков.** Срезы. Методы. Изменение элементов, вычисление суммы, произведения, поиск экстремума, сортировка. Решение задач.

**Тема 13. Обработка двумерных списков.** Обработка и вывод вложенных списков. Ввод двумерного массива. Пример обработки двумерного массива. Вложенные генераторы двумерных массивов.

**Тема 14. Отладка программ.**

**Тема 15. Работа с файлами.** Понятие файла. Классификация файлов. Назначение, открытие и закрытие файлов. Обработка текстовых файлов. Обработка типизированных файлов. Файловый ввод-вывод. Чтение данных из файла. Вывод данных в файл. Закрытие файла.

**Тема 16. Множества.** Задание множеств. Работа с элементами множеств. Перебор элементов множества. Операции с множествами.

**Тема 17. Структуры данных: словари.** Словарь можно охарактеризовать как изменяемый неупорядоченный набор пар "ключ : значение". Когда нужно использовать словари. Создание словаря. Работа с элементами словаря. Перебор элементов словаря.

### Учебный (тематический) план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации (контроля)
		Всего	Теория	Практика	
1	Участие в организационном собрании ШНО	2	2	0	
2	Введение в язык Python. Компиляторы и интерпретаторы. Правила записи команд.	2	1	1	
3	Типы данных	2	1	1	
4	Арифметические операции, целочисленная арифметика	3	1	2	Зачетные Задачи
5	Условная инструкция	3	1	2	Зачетные Задачи
6	Целочисленная арифметика и Условная инструкция	3	1	2	Контроль в тестирующей системе
7	Циклы for	2	1	1	Зачетные Задачи
8	Циклы while	2	1	1	Зачетные Задачи
9	Подготовка к Муниципальному туру ВОШ	4	2	2	Участие в Олимпиаде

10	Циклы for и while	3	1	2	Контроль в тестирующей системе
11	Действительные (вещественные) числа	3	1	2	Контроль в тестирующей системе
12	Таблица кодов ASCII	2	1	1	
13	Библиотека Math Функции ord и chr	2	1	1	Зачетные Задачи
14	Действительные (вещественные) числа	3	1	2	Контроль в тестирующей системе
15	Строки. Срезы	3	1	2	
16	Строки. Срезы. Методы.	3	1	2	Контроль в тестирующей системе
17	Функции.	2	1	1	Контроль в тестирующей системе
18	Рекурсия	2	1	1	Зачетные Задачи
19	Сложные Задачи с использованием рекурсии.	3	1	2	Контроль в тестирующей системе
20	Списки.	3	1	2	Зачетные Задачи
21	Списки и Срезы.	3	1	2	Контроль в тестирующей системе
22	Арифметические задачи.	3	1	2	Контроль в тестирующей системе
23	Сортировки.	4	2	2	Контроль в тестирующей системе
24	Символы и строки.	3	1	2	Контроль в тестирующей системе
25	Двумерные списки.	2	1	1	Зачетные Задачи
26	Генераторы двумерных списков.	3	1	2	Контроль в тестирующей системе
27	Разбор Олимпиадных задач.	5	1	4	
28	Работа с файлами.	3	1	2	Контроль в тестирующей системе
29	Словари и множества.	3	1	2	Контроль в тестирующей системе
30	Решение Олимпиадных задач.	7	2	5	Зачетные Задачи

31	Решение задач для поступления в ЛКШ (Летняя компьютерная школа).	8	2	6	Участие в Олимпиаде
32	Решение Олимпиадных задач.	9	3	6	Зачетные Задачи
	<b>Итого:</b>	<b>105 ч</b>	<b>39 ч</b>	<b>66 ч</b>	

### Организационно-педагогические условия реализации программы

Занятия должны проводиться в компьютерном классе. Где для каждого обучающегося оборудовано рабочее место, персональный компьютер с операционной системой Windows или Linux. На компьютерах должно быть установлено следующее программное обеспечение:

1. Консольный интерпретатор языка. Необходимо скачать дистрибутив с официального сайта <http://www.python.org>.
2. Система программирования IDLE – стандартная среда разработки для Python
3. Рекомендуются среда Wing IDE 101 – простая кроссплатформенная бесплатная среда для обучения. Ссылка на официальный сайт <http://www.wingware.com>
4. Возможно использование других систем программирования языка Python.
5. У каждого обучающегося должен быть организован доступ к сети Интернет для регистрации и дальнейшей работе в системе дистанционной подготовки по информатике.
6. Для школьников регистрация осуществляет ссылке: <http://informatics.mccme.ru/login/signup.php>

Для объяснения материала используются видеолекции и теоретический материал курса: **Программирование на языке Python (школа 179 г. Москвы)**. Автор: **Кириенко Д.П.**

<http://informatics.mccme.ru/course/view.php?id=156>

Программирование – это не только набор навыков практических действий, но и понимание смысла этих действий. Взаимосвязь двух сторон обучения, теории и практики, способствует развитию знаний, умений и навыков на занятиях и формирует устойчивый интерес к овладению программированием. Поэтому в связи с поставленными задачами и имеющимся оборудованием организация занятий строится следующим образом:

- Объяснение
- Показ
- Практическое закрепление полученных знаний за компьютером

Занятие должно проходить в атмосфере конструктивного взаимодействия, должен присутствовать постоянный анализ собственной деятельности, учащиеся постоянно должны получать консультации преподавателя. Обязательно должно присутствовать обсуждение итогов занятия.

Занятия должны строиться с учетом индивидуальных особенностей развития каждого ученика. Также должно учитываться наличие, или отсутствие начальных навыков работы с компьютером, а также наличие, или отсутствие у учащегося своего домашнего ПК.

Различная начальная подготовленность участников кружка требует четкого дифференцированного подхода к итогам их работы. Поэтому успешная деятельность начинающих заслуживает одобрения так же, как и успехи учащихся уже имеющих навыки обращения с компьютером.

### **Список литературы**

1. Уоррен Сэнд, Карен Сенд. Hello World. Занимательное программирование. – Спб.: Питер, 2017.

Интернет ресурсы:

2. <http://progbook.net/python/7043-izuchaem-python-4-e-izdanie.html>

Марк Лутц. Изучаем Python. 4-е издание

3. <http://progbook.ru/python/723-sammerfild-programmirovanie-na-python-3.html>

Марк Саммерфилд. Программирование на Python 3.

4. <http://progbook.net/python/6601-python-podrobnyj-spravochnik.html>

Дэвид Бизли. Python. Подробный справочник.