|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Муниципальное автономное общеобразовательное учреждение города Новосибирска «Средняя общеобразовательная школа № 213 «Открытие»

|  |  |
| --- | --- |
| СОГЛАСОВАНОЗаместитель директора по УВР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_от «\_\_\_» \_\_\_\_\_\_ 2018 | УТВЕРЖДАЮДиректор МАОУ СОШ № 213 «Открытие»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Д. ШмаковаПриказ от \_\_\_\_\_\_\_\_\_\_ №\_\_\_\_\_\_ |

**Рабочая программа** «Алгоритмизация и программирование» 10-11 классинформационно-технологический Количество часов по учебному плану:

|  |  |  |
| --- | --- | --- |
|  | 10 класс | 11 класс |
| в год | 36 | 34 |
| в неделю | 1 | 1 |

Программа составлена в соответствии с ФК ГОС Разработчики программы: Раков Андрей Егорович, Докина Анна Николаевна Новосибирск, 2018 |

**1. Пояснительная записка**

***1.1. Обоснование актуальности курса:***

Данный курс «Алгоритмизация и программирование» является расширением курса «Информатика и ИКТ». Содержание обучения, представленное в программе, позволяет вести обучение школьников в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов — блок-схем, алгоритмов, программ — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обусловливают сознательное отношение к изучению основных теоретических положений информатики.

***1.2. Цели и задачи курса***

* понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
* овладеть базовыми понятиями теории алгоритмов;
* научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

**Задачи курса:**

· познакомить с понятиями алгоритма, вычислимой функции, языка программирования;

· научить составлять и читать блок-схемы;

· сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;

· изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);

· научить применять функции при написании программ на языке программирования Python;

· научить отлаживать и тестировать программы, делать выводы о работе этих программ.

***1.3. Нормативные документы***

 Предлагаемая программа составлена на основе учебного плана 10 информационно-технологического класса на 2018-2020 учебные года МАОУ СОШ №213 «Открытие», федерального компонента государственного стандарта углубленного уровня общего образования (2004 г.), Примерной программы среднего общего образования по информатике и ИКТ. Также разработка программы осуществлена с учетом методических особенностей авторской программы к учебнику «Информатика. 10-11 классы. Углублённый уровень» [К.Ю. Полякова](http://kpolyakov.spb.ru/dosie.htm) и [Е.А. Еремина](http://pspu.ru/personal/eremin/).

***1.4. Особенности программы***

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования.

***1.5. Место программы в образовательном процессе***

Курс «Алгоритмизация и программирование» является курсом по выбору, представлен в учебном плане в компоненте образовательного учреждения и предназначен для реализации профильного обучения, расширяет и углубляет основной курс «Информатика и ИУКТ». Для полного освоения программы углубленного уровня рекомендуется изучение предмета «Информатика» по 4+1 час в неделю в 10 и 11 классах (10 классе – 144+**36** часа, 11 классе – 136+**34** часов).

***1.6. Контингент учащихся*** 10-11 классов мотивирован на изучение данного предмета. Уровень подготовки учащихся на начало обучения основан на ранее пройденной программе 7-9 класса. Курс рекомендован учащимся 10-11-х классов старшей школы, сдающих ЕГЭ по информатике.

***1.7. Применяемые методы и технологии***

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала учителем, в переходе от репродуктивного вида работ к самостоятельным, поисковоисследовательским [видам деятельности.](http://pandia.ru/text/category/vidi_deyatelmznosti/) Поэтому основная методическая установка в данном курсе — обучение учащихся навыкам самостоятельной творческой деятельности. Организация учебного процесса предусматривает дистанционной формы деятельности, когда учащийся вне уроков самостоятельно выполняет на компьютере практические задания.

***1.8. Требования к уровню подготовки учащегося***

В рамках курса «Основы программирования 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.

***1.9. Условием эффективной реализации данной программы***является наличие технического оснащения кабинета информатики:

* рабочих мест ученика
* рабочего места учителя
* проектора
* интерактивной доски
* набора программного обеспечения
	+ операционная система Windows 2016
	+ офисный пакет Microsoft Office 2016
	+ архиватор WinRar
* Среда разработки: Python 3.3 (или выше),
* наличие подключения всех учебных мест к сети Интернет

#

# Содержание

# Тема 1. Знакомство с языком Python

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:

* понятие программы;
* структура программ;
* режимы работы.

Учащиеся должны уметь:

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

# Тема 2. Переменные и выражения

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;
* операторы ввода-вывода.

Учащиеся должны уметь:

* пользоваться интерфейсом среды программирования Python;
* использовать команды редактора;
* организовывать ввод и вывод данных;
* записывать арифметические выражения.

# Тема 3. Условные предложения

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.

# Тема 4. Циклы

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 :

* 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;

# Тема 5. Функции

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач c 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.

# Тема 6. Строки - последовательности символов

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.

# Тема 7. Сложные типы данных

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:

- 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.

 **Тема 8. Стиль программирования и отладка**

**программ** 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.

**Учебно-тематический план**

|  |  |  |
| --- | --- | --- |
| **Номер урока**  | **Название**  | **Количество** **часов**  |
|  | **Тема 1. Знакомство с языком Python** | **2** |
| 1 | Общие сведения о языке | 1 |
|  | Практическая работа 1.1. Установка программы Python |  |
| 2 | Режимы работы | 1 |
|  | Практическая работа 1.2. Режимы работы с Python |  |
|  | Тест № 1. Знакомство с языком Python |  |
|  | **Тема 2. Переменные и выражения** | **4** |
| 3 | Переменные | 1 |
|  | Практическая работа 2.1. Работа со справочной системой |  |
|  | Практическая работа 2.2. Переменные |  |
| 4 | Выражения | 1 |
|  | Практическая работа 2.3. Выражения |  |
| 5 | Ввод и вывод | 1 |
| 6 | Задачи на элементарные действия с числами | 1 |
|  | Практическая работа 2.5. Задачи на элементарные действия с числами |  |
|  | Тест № 2. Выражения и операции. |  |
|  | **Тема 3. Условные предложения** | **5** |
| 7 | Логические выражения и операторы | 1 |
|  | Практическая работа 3.1. Логические выражения |  |
| 8 | Условный оператор | 1 |
|  | Практическая работа 3.2. "Условный оператор" |  |
| 9 | Множественное ветвление | 1 |
|  | Практическая работа 3.3. Множественное ветвление |  |
| 10 | Реализация ветвления в языке Python | 1 |
|  | Практическая работа 3.4. "Условные операторы" |  |
|  | Самостоятельная работа № 1 по теме "Условные операторы". |  |
| 11 | Зачетная работа № 1. "Составление программ с ветвлением". | 1 |
|  | Тест № 3. "Условные операторы". |  |
|  | **Тема 4. Циклы** | **7** |
| 12 | Оператор цикла с условием | 1 |
|  | Практическая работа 4.1. "Числа Фибоначчи" |  |
| 13 | Оператор цикла for | 1 |
|  | Практическая работа 4.2. Решение задачи с циклом for. |  |
| 14 | Вложенные циклы | 1 |
|  | Практическая работа 4.3. Реализация циклических алгоритмов |  |
| 15 | Случайные числа | 1 |
|  | Практическая работа 4.4. Случайные числа |  |
| 16 | Примеры решения задач с циклом | 1 |
|  | Практическая работа 4.5. Решение задач с циклом. |  |
|  | Самостоятельная работа № 2 "Составление программ с циклом" |  |
|  | Тест № 4. Циклы |  |
| 17 | Творческая работа № 1. "Циклы" |  |
|  | **Тема 5. Функции** | **5** |
| 18 | Создание функций | 1 |
|  | Практическая работа 5.1. Создание функций |  |
| 19 | Локальные переменные | 1 |
|  | Практическая работа 5.2. Локальные переменные |  |
| 20 | Примеры решения задач c использованием функций | 1 |
|  | Практическая работа 5.3. Решение задач с использованием функций |  |
|  | Самостоятельная работа № 3 по теме "Функции" |  |
| 21 | Рекурсивные функции | 1 |
|  | Практическая работа 5.4. Рекурсивные функции |  |
|  | Тест № 5. Функции |  |
|  | **Тема 6. Строки - последовательности символов** | **3** |
| 22 | Строки | 1 |
| 23 | Срезы строк | 1 |
|  | Практическая работа 6.1. Строки |  |
| 24 | Примеры решения задач со строками | 1 |
|  | Практическая работа 6.2. Решение задач со строками. |  |
|  | **Тема 7. Сложные типы данных** | **6** |
| 25 | Списки | 1 |
| 26 | Срезы списков | 1 |
|  | Практическая работа 7.1. Списки |  |
| 27 | Списки: примеры решения задач | 1 |
|  | Практическая работа 7.2. Решение задач со списками |  |
| 28 | Матрицы | 1 |
|  | Тест № 7. Списки |  |
| 29 | Кортежи | 1 |
| 30 | Введение в словари | 1 |
| 31 | Множества в языке Python | 1 |
|  | **Тема 8. Стиль программирования и отладка программ** | **5** |
| 32 | Стиль программирования | 1 |
| 33 | Отладка программ | 1 |
| 34 | Зачет по курсу «Программирование на языке Python» | 1 |
| 35 | Подведение итогов | 1 |
| 36 | Подведение итогов  | 1 |
|  | **ВСЕГО**  | **36**  |

# Литература и источники

1. Домашняя страница Python [www.python.org](http://www.python.org/) . Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс [«Введение в программирование на Python»](http://www.intuit.ru/studies/courses/12179/1172/info), [http://www.intuit.ru/studies/courses/12179/1172/info.](http://www.intuit.ru/studies/courses/12179/1172/info)
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования *Python*» [http://www.intuit.ru/studies/courses/49/49/info.](http://www.intuit.ru/studies/courses/49/49/info)
4. Сайт проекта Open Book Project [openbookproject.net](http://openbookproject.net/) содержит серию практических примеров на Python Криса Мейерса.
5. *Python. Подробный справочник* Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
6. *Python. Справочник* Марка Лутца. Справочник по наиболее часто использующимся функциям и модулям.