

АНО ДО «Центр развития «ЭВРИКА»

ЭВРИКА
центр
развития

«Согласовано»

Проректор по учебной работе КемГУ

Р.М. Котов
«__» _____ 2018



«Утверждено»

Директор АНО ДО «Центр развития
«Эврика»

Л.К. Асякина
«__» _____ 2018



РАБОЧАЯ ПРОГРАММА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

«Школа программирования (на языке Python)»

5-11 класс

Автор-составитель:

директор АНО ДО «Центр развития «Эврика»

Асякина Л.К.

Кемерово – 2018

Содержание

1. Результаты освоения курса	3
2. Содержание курса внеурочной деятельности с указанием форм организации и видов деятельности	3
3. Учебный план	6

1. Результаты освоения курса внеурочной деятельности

Личностные:

- сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
- осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов.

Предметные:

- владение навыками 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;
- владение навыками и опытом разработки программ на Python, включая тестирование и отладку программ;

Метапредметные:

- 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.

2. Содержание курса внеурочной деятельности с указанием форм организации и видов деятельности

Изучение темы алгоритмизации и основ программирования в курсе информатики является одним из самых сложных в рамках предмета. Учитывая относительно небольшой объём учебного времени, выделяемый на этот блок, а также то, что для большинства учащихся самостоятельное изучение принципов алгоритмизации и программирования по учебникам и учебным пособиям затруднительно, перед учителем информатики встаёт проблема разработки таких способов и методик, которые помогут детям глубже понять суть алгоритмизации, научиться программировать на одном или нескольких языках, возможно, помочь определиться старшеклассникам в профессиональном плане.

Одной из целей создания языка Python считается обучение студентов структурному программированию. До сих пор Python заслуженно считается одним из лучших языков для начального обучения программированию.

Содержание программы включает поддержку трёх основных аспектов преподавания информатики в школе:

«Пользовательский» аспект, связанный с формированием компьютерной грамотности, информационной культуры, подготовкой

школьников к практической деятельности в условиях широкого использования информационных технологий.

Алгоритмический (программистский) аспект, связанный с развитием алгоритмического стиля мышления учащихся.

Кибернетический аспект, связанный с формированием мировоззренческих представлений о роли информации в управлении, закономерностей информационных процессов.

В основе программы осуществляется интегрированный подход к изучению тем, что позволяет учащимся глубже овладеть необходимыми знаниями. Интегрирование ведётся по трём направлениям: математика, физика и информатика.

Актуальность программы.

В наше время важно, чтобы человек не только умел работать за компьютером, но и понимал, как устроены программы, с помощью которых он работает на нём. Обучение по программе станет стартовой базой для изучения языка объектно-ориентированного программирования. Занятия помогут учащимся глубже изучить один из языков программирования (в нашем случае, Python) и приобрести необходимые навыки в составлении программ. Для учащихся, которые имеют склонность быстрее осваивать основные принципы программирования, можно давать индивидуальные задания олимпиадного характера.

Требования высших учебных заведений к выпускникам школ существенно возросли и расширились, поэтому целью также является оказание помощи в изучении основ программирования и подготовке учащихся к поступлению и успешному обучению в ВУЗах технической направленности.

Цели программы.

Закрепить у учащихся понятия алгоритма, свойства алгоритмов, способы записей алгоритмов, основных алгоритмических структур (линейной, ветвления, цикла), вспомогательных алгоритмов.

Обратить особое внимание на алгоритмическое программирование: основные типы и структуры данных (переменные, массивы), процедуры и функции.

Дать учащимся представление о решении сложных задач

Задачи программы.

• Обучающие

Расширить знания учащихся по алгоритмизации и программированию.

Привить учащимся навыки решения простых задач по программированию.

Научить решать сложные задачи методом деления на подзадачи.

• Развивающие

Программа рассчитана на развитие познавательных способностей учащихся, углубление интереса к программированию, привитие

алгоритмического стиля мышления. Программа рассчитана также на подготовку некоторых учащихся к олимпиадам по программированию.

- **Воспитательные**

Программа рассчитана на воспитание у учащихся чувства ответственности, анализа ситуации и поиска мер по её разрешению. Составление алгоритмов и программ позволяет ученику оценивать задачу и заняться поиском вариантов её решения, что позволит ему и в других жизненных ситуациях сделать то же самое. Работа в малых группах способствует развитию коммуникативных способностей, взаимовыручке, умению выработать единый подход к решению задачи.

После изучения курса учащиеся должны:

- знать место языка Python среди языков программирования высокого уровня,
- знать особенности структуры программы, представленной на языке Python,
- иметь представление о модулях, входящих в состав среды Python.
- знать возможности и ограничения использования готовых модулей.
- иметь представление о величине, ее характеристиках,
- знать что такое операция, операнд и их характеристики,
- знать принципиальные отличия величин структурированных и не структурированных,
- иметь представление о таких структурах данных, как число, текст, кортеж, список, словарь,
- иметь представление о составе арифметического выражения;
- знать математические функции, входящие в Python,
- иметь представление о логических выражениях и входящих в них операндах, операциях и функциях,
- уметь записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить,
- знать основные операторы языка Python, их синтаксис,
- иметь представление о процессе исполнения каждого из операторов,
- уметь разрабатывать программы обработки числовой и символьной информации,
- уметь разрабатывать программы (линейные, разветвляющиеся и с циклами),
- иметь представление о значении полноценных процедур и функций для структурно-ориентированного языка высокого уровня,
- знать правила описания процедур и функций в Python и построение вызова процедуры,
- знать принципиальные отличия между формальными, локальными и глобальными переменными,
- знать область действия описаний в процедурах,
- иметь представление о рекурсии, знать ее реализацию на Python,
- владеть основными приемами формирования процедуры и функции,

- знать, как с помощью списков определять в программе тип «массив», «матрица»
- знать свойства данных типа «массив», «матрица»
- уметь воспроизводить алгоритмы сортировки массивов и матриц, поиска в упорядоченном массиве, распространять эти алгоритмы на сортировку и поиск в нечисловых массивах
- уметь читать и записывать текстовые файлы в заданном формате.
- решать основные алгоритмические задачи, представленные в пояснительной записке, в среде Python.

3. Учебный план

История языков программирования. Компиляция и интерпретация.

Знакомство с Python и средами программирования.

Типы данных в программировании. Определение переменной.

Ввод данных с клавиатуры.

Логические выражения.

Условный оператор. Инструкция if.

Множественное ветвление.

Цикл While и For.

Строки как последовательности символов.

Списки — изменяемые последовательности. Массивы.

Основные задачи обработки массивов: поиск, сортировка, реверс, ...

Введение в словари.

Функции в программировании. Параметры и аргументы функций.

Локальные и глобальные переменные. Процедуры.

Файлы. Чтение текстового файла. Запись в файл.

Алгоритм Евклида (нахождение наибольшего общего делителя)

Вычисление факториала на языке программирования Python

Двоичный (бинарный) поиск элемента в массиве

Замена элементов в списке

Перевод чисел из десятичной системы счисления в двоичную

Решето Эратосфена - алгоритм определения простых чисел

Сортировка выбором (поиск минимума и перестановка)

Сортировка методом пузырька

Сумма и произведение цифр числа

Тестирование простоты числа методом перебора делителей

Числа Фибоначчи (вычисление с помощью цикла while и рекурсии)