Муниципальное общеобразовательное учреждение Белоярского района

«Средняя общеобразовательная школа № 2 г. Белоярский»

СОГЛАСОВАНО	УТВЕРЖДЕНО
заместитель директора по УВР	директор
Истомина О.Ю.	Исаченко Н.В.
	Приказ от 30.08.2024 г. № 300

Программа дополнительного образования

Язык программирования Python

для обучающихся 8-9 классов

срок реализации программы – 1 год

автор программы: учитель информатики

Карауловская Ольга Евгеньевна

г. Белоярский, 2024

Пояснительная записка

Место курса в образовательном процессе

Изменение взглядов на предмет информатики как науки, её место в системе научного знания требует существенных изменений в содержании образования по информатике. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей среды.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей, связанных с познанием, если они видят смысл в сотрудничестве с одноклассниками и учителем.

Содержание обучения, представленное в программе курса «Основы алгоритмизации и программирование на языке Python», позволяет вести обучение школьников в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов — блок-схем, алгоритмов, исполняемых файлов — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики.

Проявления трудолюбия, целеустремлённости и одухотворённости, возникающие при воплощении замыслов учащихся в рамках курса «Основы алгоритмизации и программирование на языке Python», стимулируют развитие индивидуально-личностных качеств школьников.

Здесь они научатся работать с основными типами данных и операторов, познакомятся с построением графических изображений средствами языка Python.

Активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует

созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся.

Общие требования к образованности учащихся

Для качественного обучения необходимо, чтобы учащиеся обладали базовыми знаниями по математике на уровне 7 класса и навыками работы в файловой системе Windows.

Также приветствуется умение учиться независимо от других, планировать и организовывать свою деятельность.

Концепция курса

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.

Для школьников, выбравших информационно-технологический профиль, этот курс — возможность развить навыки программирования на языке Python. Программирование — это стержень как базового, так и профильного курсов информатики. В рамках предлагаемого курса «Основы алгоритмизации и программирование на языке Python» изучение основ программирования на языке Python — это не столько средство подготовки к будущей профессиональной деятельности, 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.

Общепедагогическая направленность занятий — гармонизация индивидуальных и социальных аспектов обучения по отношению к информационным технологиям. 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. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается

основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Цели изучения курса

Основными целями курса являются:

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- овладеть базовыми понятиями теории алгоритмов;
- освоить понятие алгоритма и особенности реализации алгоритмов в виде программ, написанных на языке программирования Python.

Задачи курса

Основными задачами курса являются:

- познакомиться с понятиями «алгоритм», «язык программирования»;
- научиться составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python;
- научиться работать с графическими средствами языка программирования Python;
- научиться отлаживать и тестировать программы, делать выводы о работе этих программ.

Методы обучения

Отбор методов обучения обусловлен необходимостью формировать у старшеклассников информационную и коммуникативную компетентности, 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, поисково-исследовательским видам деятельности. Поэтому основным методом обучения в данном элективном курсе является метод проектов, а основная методическая установка —

обучение старшеклассников навыкам самостоятельной творческой деятельности.

Формы организации учебных занятий

Организация учебного процесса с использованием учебно-методического комплекта предусматривает наличие двух взаимосвязанных и взаимодополняющих форм:

- *урочная форма*, когда учитель во время урока консультирует учащихся в процессе выполнения ими практических заданий на компьютере;
- *внеурочная форм*а, когда учащийся вне уроков самостоятельно выполняет на компьютере практические задания.

Планируемые результаты курса

В рамках курса «Основы алгоритмизации и 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 алгоритмизации и программирование 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.

Педагогическая ценность контроля заключается в том, что он даёт всестороннюю информацию о способностях учащихся к анализу или синтезу, оценочным суждениям и позволяет оценить эффективность учебного труда для каждого из них.

Диагностика и контроль — необходимые части учебного процесса, но увеличение их доли неизбежно приводит к сокращению времени на изучение материала. Поэтому столь важно извлечение максимума информации об учащихся за минимальное время. Контроль и диагностика должны быть действенными. Поэтому необходимо анализировать результаты проверки и принимать меры по коррекции образовательного процесса. От этого зависит, станут ли способы оценивания уровня достижений учащихся результативными.

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.).

Созданными внешними образовательными продуктами учащиеся могут пополнять собственные портфолио.

Проверка достигаемых школьниками результатов производится в следующих формах:

- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка учителем деятельности школьников в виде трёх контрольных работ по следующим темам: «Алгоритмизация. Знакомство с Python»; «Основные алгоритмические конструкции»; «Структурированные типы данных».

Итоговый контроль проводится в конце курса. Он организуется в форме экзамена — защита итогового проекта.

Состав учебно-методического комплекта

Программа курса обеспечивается учебным пособием для учащихся «Основы алгоритмизации и программирование на языке Python», электронным практикумом «Уроки программирования», контрольно-измерительными материалами для проведения текущего контроля и экзамена.

Курс, имея собственную доминантную направленность, предполагает интеграцию с другими учебными предметами. Информационная составляющая этих предметов может использоваться школьниками в процессе разработки программ.

Аппаратное обеспечение:

1. IBM PC-совместимый компьютер.
2. Процессор не ниже Pentium-100.
3. Оперативная память не меньше 64 Мб.

Программное обеспечение:

4. Операционная система: Windows XP (или выше).
5. Одна из сред разработки

Тематический план курса

Наименование разделов и тем	Количество часов	
	Всего	**Прак. занятия**
Раздел 1. Основы алгоритмизации	**7**	**7**
Раздел 2. Знакомство с Python	**14**	**14**
2.1. Структура программы на языке Python	2	2
2.2. Числовые типы данных	4	4
2.3. Подпрограммы	2	2
2.4. Использование графического модуля	4	4
Контрольная работа № 1	2	2
Раздел 3. Основные алгоритмические конструкции	**24**	**24**
3.1. Циклы	6	6
3.2. Условный оператор. Оператор выбора	6	6
3.3. Средства отладки программ	4	4
3.4. Константы	2	2
3.5. Компьютерная анимация	4	4
Контрольная работа № 2	2	2
Раздел 4. Структурированные типы данных	**22**	**22**
4.1. Массивы	6	6
4.2. Типизированные константы	2	2
4.3. Строковый тип данных	4	4
4.4. Записи	2	2
4.5. Файловый тип данных	6	6
Контрольная работа № 3	2	2
Экзамен	**2**	**2**
Резерв времени	2	
ВСЕГО	**72**	**69**

Содержание курса

Раздел 1. Основы алгоритмизации

Учащиеся должны знать / понимать:

- понятие алгоритма;
- понятие исполнителя;
- назначение и основные команды среды исполнителя;

- типы алгоритмов;
- свойства алгоритма;
- язык блок-схем.

Учащиеся должны уметь:

- составлять несложные алгоритмы для исполнителя;
- записывать алгоритм разными способами;
- определять исполнителя алгоритма.

Алгоритмы. Способы записи алгоритма. Исполнители алгоритмов. Типы алгоритмов: вспомогательные, циклические, разветвляющиеся. Определение и свойства алгоритма.

Практическая работа: создание различных алгоритмов для исполнителя Кенгурёнок. Уроки электронного практикума:

Урок 1 «Кубики, ступеньки, или Знакомство с исполнителем».

Урок 2 «Тетрадь в линеечку, или Циклический алгоритм».

Урок 3 «Умный в гору не пойдёт, или Ветвление».

Урок 4 «Независимое расследование, или Что же такое алгоритм».

Раздел 2. Знакомство с Python

Тема 2.1. Структура программы на языке Python

Учащиеся должны знать / понимать:

- назначение и основные команды среды разработки;
- общую структуру программы;
- назначение и виды оператора вывода.

Учащиеся должны уметь:

- пользоваться интерфейсом среды программирования Borland / Turbo Python или Free Python
- использовать команды редактора;
- составлять и запускать программы;
- организовывать вывод данных.

Язык программирования Python и его характерные особенности. Структура программы на языке Python. Простейшая программа. Среды разработки. Элементы языка Python. Создание и исполнение программ в среде разработки. Операторы вывода Write и WriteLn.

Практическая работа: создание, сохранение, запуск простейшей программы в среде разработки. Урок 5 электронного практикума «Hello, World! или Знакомство с Python».

Тема 2.2. Числовые типы данных

Учащиеся должны знать / понимать:

- понятие типа данных;

- целые, вещественные типы данных и операции над ними;
- понятие переменной;
- оператор присваивания;
- назначение и виды оператора ввода.

Учащиеся должны уметь:

- определять тип числовых данных;
- объявлять необходимые переменные;
- записывать арифметические выражения.

Переменные. Типы данных в языке Python. Простые типы данных. Целые и вещественные типы. Значения. Оператор присваивания. Операции, допустимые с переменными и значениями целого и вещественного типа.

Практическая работа: составление вычислительных программ. Уроки электронного практикума:

Урок 6 «Посчитаем, или Типы данных».

Урок 7 «Решаем уравнения, или Вычисления в программе».

Тема 2.3. Подпрограммы

Учащиеся должны знать / понимать:

- назначение подпрограмм;
- отличия процедур и функций;
- понятие формальных и фактических параметров.

Учащиеся должны уметь:

- объявлять процедуры и функции в программе Python;
- вызывать подпрограммы из основной программы.

Подпрограмма. Процедуры и функции. Параметры, формальные и фактические параметры.

Практическая работа: создание программ с использованием различных видов подпрограмм. Урок 8 электронного практикума «И снова уравнение, или Подпрограммы».

Тема 2.4. Использование графического модуля

Учащиеся должны знать / понимать:

- понятие модуля;
- назначение и возможности графического модуля.

Учащиеся должны уметь:

- подключить графический модуль;
- инициализировать графический режим;
- использовать графические примитивы.

Модули. Модуль **Graph**, назначение и возможности. Графический экран (режим). Основные графические примитивы. Управление цветом. Штриховка.

Практическая работа: создание графических программ. Уроки электронного практикума:

Урок 9 «Нарисуем — будем жить, или Графический модуль».

Урок 10 «Белокрылые лошадки, или Относительные координаты».

Контрольная работа № 1.

Раздел 3. Основные алгоритмические конструкции

Тема 3.1. Циклы

Учащиеся должны знать / понимать:

- понятие и назначение цикла;
- цикл со счётчиком;
- циклы с условием;
- понятие генератора случайных чисел;
- понятие символьного типа;
- назначение и возможности модуля **CRT**;
- понятие кода клавиши, расширенного кода клавиши.

Учащиеся должны уметь:

- использовать все виды циклов для повторения блока действий в программе;
- определять оптимальный вид оператора цикла для решения поставленной задачи;
- использовать генератор случайных чисел;
- использовать символьные переменные и константы;
- принимать коды и расширенные коды клавиш: символьных и служебных.

Цикл. Цикл со счётчиком. Цикл с предусловием и цикл с постусловием.

Генератор случайных чисел.

Символьный тип данных Использование возможностей модуля CRT для приёма и обработки сигналов клавиш.

Практическая работа: создание программ, использующих разные виды циклов. Уроки электронного практикума:

Урок 11 «И получилась звёздная дорога, или Цикл с параметром».

Урок 15 «Змейкой петляет в ущелье дорога, или Цикл repeat-until».

Урок 16 «Алгоритм Евклида, или Сравниваем циклы».

Тема 3.2. Условный оператор. Оператор выбора

Учащиеся должны знать / понимать:

- понятие и назначение условного оператора;
- назначение оператора выбора;
- алгоритм поиска максимального/минимального элемента;
- логические выражения;
- способы тестирования программ.

Учащиеся должны уметь:

- использовать условный оператор, оператор выбора при составлении программ;
- осуществлять выбор типа условного оператора/оператора выбора для оптимального решения поставленной задачи;
- составлять сложные логические выражения;
- использовать алгоритм поиска максимального/минимального элемента последовательности;
- составлять тестовую таблицу, тестировать готовую программу.

Условный оператор. Полная и неполная формы условного оператора. Оператор выбора.

Алгоритм поиска максимального / минимального элемента последовательности.

Тестирование готовой программы.

Практическая работа: создание программ, использующих алгоритмы ветвления. Уроки электронного практикума:

Урок 12 «С неба звёздочку достану, или Условный оператор».

Урок 13 «Самый высокий, или Полная форма условного оператора».

Урок 14 «Ти ж мене пидманула, или Оператор выбора».

Тема 3.3. Средства отладки программ

Учащиеся должны знать / понимать:

- механизм отладки;
- возможности отслеживания значений переменных;
- способы пошагового выполнения программы;
- метод дихотомии.

Учащиеся должны уметь:

- вывести в окно отладки имена переменных;
- произвести пошаговое выполнение программы;
- тестировать программу, выявлять и исправлять ошибки;
- находить корни произвольных уравнений методом дихотомии.

Отладка. Окно **`Watches`**. Пошаговое выполнение программы.

Алгоритм обмена значений двух переменных. Поиск корней уравнения методом дихотомии.

Практическая работа: отладка программы, поиск логических ошибок. Уроки электронного практикума:

Урок 17 «Ты — мне, я — тебе, или Работа с отладчиком».

Урок 18 «Всё поделим пополам, или Метод дихотомии».

Тема 3.4. Константы

Учащиеся должны знать / понимать:

- понятие константы;
- принципы преобразования экранных координат.

Учащиеся должны уметь:

- объявить константу, использовать её значение в программе;
- строить график произвольной функции в заданном масштабе и расположении.

Константы. Объявление константы. Использование константы в программе.

Преобразования экранных координат. Построение графика функции.

Практическая работа: создание программы для построения графика функции. Урок 19 электронного практикума «Преобразования экранных координат, или Константы».

Тема 3.5. Компьютерная анимация

Учащиеся должны знать / понимать:

- принципы компьютерной анимации;
- назначение и отличия функций **`ReadKey`** и **`KeyPressed`**.

Учащиеся должны уметь:

- реализовать несложное движение объектов на экране, в том числе и с переменной формой.

Компьютерная анимация. Передвижение объекта по заданной траектории. Передвижение объекта с изменяющейся формой.

Операции инкремента и декремента, их преимущества перед оператором присваивания.

Организация задержки в программе с помощью процедуры **`delay`**.

Практическая работа: создание программ, реализующих передвижение объекта по экрану. Уроки электронного практикума:

Урок 20 «Мой весёлый звонкий мяч, или Анимация».

Урок 21 «Про маленькую гордую гусеницу, или Покадровая анимация».

Контрольная работа № 2.

Раздел 4. Структурированные типы данных

Тема 4.1. Массивы

Учащиеся должны знать / понимать:

- понятие массива;
- понятие двумерного массива как массива массивов;
- способы поиска максимального/минимального элемента массива;
- способы сортировки — сортировка выбором и пузырьковая.

Учащиеся должны уметь:

- объявлять одномерные и двумерные массивы;
- использовать массивы для хранения данных в программе;
- осуществлять поиск максимального/минимального элемента в одномерном массиве;
- производить сортировку одномерного массива одним из двух способов: сортировкой выбором или пузырьковой сортировкой.

Массив. Одномерный массив. Двумерный массив. Объявление массивов. Обработка массивов: поиск элемента по заданным признакам, заполнение массива, вывод массива на экран.

Сортировка массива. Сортировка выбором. Пузырьковая сортировка.

Практическая работа: создание и обработка массивов. Уроки электронного практикума:

Урок 22 «Угол падения… или Массивы».

Урок 23 «Самый большой, или Двумерные массивы».

Урок 24 «От заката до восхода, или Сортировка».

Тема 4.2. Типизированные константы

Учащиеся должны знать / понимать:

- понятие типизированной константы;
- область применения типизированных констант;
- отличие типизированных констант от переменных и от констант;
- принципы передвижения рисованного объекта по экрану без следа.

Учащиеся должны уметь:

- объявлять типизированные константы;
- использовать массив-буфер для сохранения области экрана при передвижении объекта по экрану.

Типизированные константы.

Алгоритм передвижения объекта по экрану без оставления следа. Использование битового образа для хранения образа объекта. Использование массива-буфера для хранения области экрана.

Практическая работа: создание программы, реализующей передвижение объекта по экрану. Урок 25 электронного практикума «Какой ты за собой оставишь след, или Типизированные константы».

Тема 4.3. Строковый тип данных

Учащиеся должны знать / понимать:

- строковый тип данных;
- понятие строк как массива символов;
- допустимые действия над строковыми данными.

Учащиеся должны уметь:

- объявлять и использовать в программе величины строкового типа;
- использовать стандартные процедуры и функции обработки строк.

Строковый тип данных. Строки как массив символов. Стандартные строковые процедуры и функции.

Практическая работа: обработка данных строкового типа. Уроки электронного практикума:

Урок 26 «Разговор с попугаем, или Строковый тип данных».

Урок 27 «Шоу бегущих строк, или Этюды об одном типе данных».

Тема 4.4. Записи

Учащиеся должны знать / понимать:

- понятие типа «запись»;
- понятия «поле», «запись»;
- область применения типа «запись»;
- понятие пользовательского типа.

Учащиеся должны уметь:

- создавать пользовательский тип;
- использовать записи для хранения базы данных.

Тип **`record`**. Пользовательские типы данных. Поля записи. Поле-массив. Поле-запись. Массив записей.

Заполнение массива записей. Вывод значения записи на экран.

Практическая работа: Заполнение «базы данных» с использованием типа **`record`**. Урок 28 электронного практикума «Живут студенты весело, или Записи».

Тема 4.5. Файловый тип данных

Учащиеся должны знать / понимать:

- понятие файлового типа;
- отличия и область применения типизированных и текстовых файловых типов;
- порядок работы с данными файлового типа.

Учащиеся должны уметь:

- определить оптимальный тип файловых данных для решения конкретной задачи;
- ассоциировать файловую переменную с файлом на диске;
- открыть файл для чтения или записи;
- записать/прочитать информацию из файла.

Файловый тип данных. Типизированные файлы. Текстовые файлы.

Практическая работа: создание программ, позволяющих хранить данные на диске. Уроки электронного практикума:

Урок 29 «Заметка на память, или Типизированные файлы».

Урок 30 «Графический редактор, или Работа с текстовыми файлами».

Урок 31 «Текстовый редактор, или Работа с текстом на экране».

Контрольная работа № 3.

Учебно-методические материалы

1. Татарникова, Л. А. Основы алгоритмизации и программирование на языке Python: Учебное пособие.
2. Уроки программирования: Электронный практикум.
3. Основы алгоритмизации и программирование на языке Python: Учебная программа.
4. Татарникова, Л. А. Основы алгоритмизации и программирование на языке Python: Методические рекомендации.
5. Татарникова, Л. А. Основы алгоритмизации и программирование на языке Python: Задания для проведения контрольной работы № 1 «Алгоритмизация. Знакомство с Python».
6. Татарникова, Л. А. Основы алгоритмизации и программирование на языке Python: Задания для проведения контрольной работы № 2 «Основные алгоритмические конструкции».
7. Татарникова, Л. А. Основы алгоритмизации и программирование на языке Python: Задания для проведения контрольной работы № 3 «Структурированные типы данных».
8. Татарникова, Л. А. Основы алгоритмизации и программирование на языке Python: Задание к итоговому проекту.