ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ГОРОДА МОСКВЫ

«ЦЕНТР СПОРТА И ОБРАЗОВАНИЯ «ОЛИМП»
ДЕПАРТАМЕНТА СПОРТА ГОРОДА МОСКВЫ
(ГБОУ «ЦСиО «Олимп» Москомспорта)

Согласовано:
Заместитель руководителя
общеобразовательной школы
ГБОУ «ЦСиО «Олимп» Москомспорта
Н.В. Шалимова
28.08.2019 г.

Рассмотрено
на заседании кафедры
естественно-
математического цикла
Г.А.Тупицина
27.08.2019 г.

РАБОЧАЯ ПРОГРАММА

Курс по выбору	Юный программист
Сроки реализации программы	2 года
Классы	8-9

Москва 2019 г.

Планируемые результаты освоения курса «Юный программист»

Планируемые результаты освоения обучающимися основной образовательной программы основного общего образования уточняют и конкретизируют общее понимание личностных, метапредметных и предметных результатов как с позиции организации их достижения в образовательном процессе, так и с позиции оценки достижения этих результатов.

Личностные результаты – это сформировавшаяся в образовательном процессе система ценностных отношений учащихся к себе, другим участникам образовательного процесса, самому образовательному процессу, объектам познания, результатам образовательной деятельности. Основными личностными результатами, формируемыми при изучении информатики в основной школе, являются:

- наличие представлений об информации как важнейшем стратегическом ресурсе развития личности, государства, общества;
- понимание роли информационных процессов в современном мире;
- владение первичными навыками анализа и критичной оценки получаемой информации;
- ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- развитие чувства личной ответственности за качество окружающей информационной среды;
- способность увязать учебное содержание с собственным жизненным опытом, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества;
- готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ;
- способность и готовность к общению и сотрудничеству со сверстниками и взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой деятельности;
- способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ.

Метапредметные результаты – освоенные обучающимися на базе одного, нескольких или всех учебных предметов способы деятельности, применимые как в рамках образовательного процесса, так и в других жизненных ситуациях. Основными метапредметными результатами, формируемыми при изучении информатики в основной школе, являются:

- овладение основными общеучебными умениями информационно-логического характера, например: анализ объектов и ситуаций; синтез как составление целого из частей и самостоятельное достраивание

недостающих компонентов; выбор оснований и критериев для сравнения и классификации объектов; обобщение и сравнение данных; установление причинно - следственных связей; построение логических цепочек рассуждений;

- овладение умениями организации собственной учебной деятельности, включающими: целеполагание – постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить; планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи; прогнозирование результата; контроль полученного результата (обнаружение ошибки) и коррекция плана действий в случае обнаружения ошибки; оценка – осознание учащимся того, насколько качественно им решена учебно-познавательная задача;
- овладение основными универсальными умениями информационного характера: постановка и формулирование проблемы; поиск и выделение необходимой информации; структурирование информации; выбор наиболее рациональных способов решения задач в зависимости от конкретных условий; самостоятельное создание алгоритмов деятельности для решения проблем творческого и поискового характера;
- овладение информационным моделированием как основным методом приобретения знаний: умение строить разнообразные информационные структуры для описания объектов; умение «читать» таблицы, графики, схемы; умение выбирать форму представления информации в зависимости от стоящей задачи;
- овладение начальными навыками исследовательской деятельности, проведения виртуальных экспериментов; овладение способами и методами освоения новых инструментальных средств;
- овладение основами продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми: умение правильно и однозначно сформулировать мысль в понятной собеседнику форме; умение осуществлять в коллективе совместную информационную деятельность, в частности при выполнении проекта; умение выступать перед аудиторией, представляя ей результаты своей работы с помощью средств ИКТ; использование коммуникативных технологий в учебной деятельности и повседневной жизни;

Предметные результаты включают в себя: освоенные обучающимися в ходе изучения учебного предмета умения специфические для данной предметной области, виды деятельности по получению нового знания в рамках учебного предмета, его преобразованию и применению в учебных, учебно-проектных и социально-проектных ситуациях, формирование научного типа мышления, научных представлений о ключевых теориях, типах и видах отношений, владение научной терминологией, ключевыми понятиями, методами и приемами. В соответствии с федеральным государственным образовательным стандартом общего образования основные предметные результаты изучения информатики в основной школе отражают:

- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- умение формально выполнять алгоритмы;
- умение создавать алгоритмы для управления виртуальными исполнителями;
- умение строить информационные модели объектов и процессов;
- умение составлять алгоритм решения задач;
- на их основе разрабатывать компьютерные модели с использованием языка Python 3;
- умение проводить компьютерный эксперимент, т.е. исследовать и анализировать компьютерные модели.
- умение создавать и выполнять программы для решения несложных алгоритмических задач построения графических изображений в выбранной среде программирования;
- умение использовать готовые прикладные компьютерные программы и сервисы в учебной деятельности.

Рабочая программа курса «Юный программист. Язык программирования Python 3» предназначена для учащихся 8-9 класса и нацелена на:

- развитие познавательных, интеллектуальных и творческих способностей учащихся, их образного, логического и алгоритмического мышления;
- воспитание интереса к информатике, стремления использовать полученные знания в процессе обучения другим предметам и в жизни;
- формирование обще-учебных умений и навыков на основе средств и методов информатики и ИКТ, в том числе овладение умениями работать с различными видами информации, самостоятельно планировать и осуществлять индивидуальную и коллективную информационную деятельность, представлять и оценивать ее результаты.

Данный курс позволит расширить кругозор обучающихся. Посредством формирования начальных навыков программирования готовится платформа для изучения более сложных языков. Данный учебный план позволит учащемуся, прошедшему курс обучения, самостоятельно моделировать алгоритмические конструкции. В процессе программирования формируется развитие логического мышления, вырабатывается целеустремленность в выборе будущего профиля обучения.

В любой среде программирования реализуются основные алгоритмические конструкции, развивающие алгоритмический стиль мышления, важность которого отмечена Н.М. Амосовым, Н.Н. Моисеевым, А.Н. Лонда и другими учеными. Ими подчеркивалась необходимость разработки алгоритмов для развития мышления школьников. Они показывали, что с помощью алгоритмов можно не только организовывать мыслительную деятельность, но и описывать процессы.

Алгоритмы возникают не только в ходе описания какого-либо процесса (физического, химического, биологического, математического), но и в управлении, воспитании, во всей социальной сфере жизни человека. Именно это и доказывает необходимость их введения в обучение. Таким образом, алгоритм – это не программа-шаблон, а механизм, согласно которого функционирует, развивается любая самоорганизующая система. Некоторые алгоритмы человек осваивает самостоятельно, другие требуют обучения.

Фундаментальное понятие информатики - «алгоритмизация», имеет большое значение не только в теории информатики, но и в теории самореализации в развитии ученика.

Объем минимального содержания базового курса информатики включает в себя блок «Алгоритмы и исполнители». Алгоритмизация - одно из мощных средств развития мышления учащихся.

Одно из перекрестных средств знакомства учащихся с основными алгоритмическими конструкциями является язык Python 3.

В последнее время язык программирования Python 3 завоевывает все большую популярность и не только в средней школе, но и в колледжах, университетах, да и просто в среде любителей интеллектуального досуга.

Язык программирования Python 3 — это мощный инструмент для создания программ самого разнообразного назначения, доступный даже для новичков. С его помощью можно решать задачи различных типов.

Язык Python обладает некоторыми примечательными особенностями, которые обуславливают его широкое распространение. Поэтому прежде чем изучать python, следует рассказать о его достоинствах и недостатках.

Python 3: преимущества и недостатки языка

1. Python - интерпретируемый язык программирования. С одной стороны, это позволяет значительно упростить отладку программ, с другой - обуславливает сравнительно низкую скорость выполнения.
2. Динамическая типизация. В python не надо заранее объявлять тип переменной, что очень удобно при разработке.
3. Хорошая поддержка модульности. Вы можете легко написать свой модуль и использовать его в других программах.
4. Встроенная поддержка Unicode в строках. В Python необязательно писать всё на английском языке, в программах вполне может использоваться ваш родной язык.
5. Поддержка объектно-ориентированного программирования. При этом его реализация в python является одной из самых понятных.
6. Автоматическая сборка мусора, отсутствие утечек памяти.
7. Интеграция с C/C++, если возможностей python недостаточно.
8. Понятный и лаконичный синтаксис, способствующий ясному отображению кода. Удобная система функций позволяет при грамотном подходе создавать код, в котором будет легко разобраться другому человеку в случае необходимости. Также вы сможете научиться читать программы и модули, написанные другими людьми.
9. Огромное количество модулей, как входящих в стандартную поставку Python 3, так и сторонних. В некоторых случаях для написания программы

достаточно лишь найти подходящие модули и правильно их скомбинировать. Таким образом, вы можете думать о составлении программы на более высоком уровне, работая с уже готовыми элементами, выполняющими различные действия.

10. Кроссплатформенность. Программа, написанная на Python, будет функционировать совершенно одинаково вне зависимости от того, в какой операционной системе она запущена. Отличия возникают лишь в редких случаях, и их легко заранее предусмотреть благодаря наличию подробной документации.

Поэтому целесообразно использовать этот язык при изучении информатики или технологии (работа в компьютерном классе) в среднем звене. В связи с этим становится очевидным актуальность предлагаемого курса.

Количество часов в неделю: 1 час (69 учебных часов за 2 года).

Цель курса: Освоение языка Python. Развитие навыков решения разнообразных задач, решаемых при помощи Python. Освоение алгоритмизации.

Для достижения комплекса поставленных целей необходимо решить следующие ***задачи***:

- включить в учебный процесс содержание, направленное на формирование у школьников основных обще-учебных умений информационно-логического характера;
- создать условия для овладения основными универсальными умениями информационного характера;
- сформировать у учащихся умения организации собственной учебной деятельности;
- организовать работу в виртуальных лабораториях и учебных средах, направленную на овладение первичными навыками исследовательской деятельности, получение опыта принятия решений и управления объектами с помощью составления для них алгоритмов;
- создать условия для овладения основами продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми.

Содержание курса

«Юный программист. Язык программирования Python 3»

25 PEP 8 - руководство по написанию кода на Python

25.1 Содержание.
25.2 Внешний вид кода.
25.3 Пробелы в выражениях и инструкциях.
25.4 Комментарии.
25.5 Контроль версий.
25.6 Соглашения по именованию.
25.7 Общие рекомендации.

26 Документирование кода в Python. PEP 257

26.1 Что такое строки документации?
26.2 Однострочные строки документации.
26.3 Многострочные строки документации.

27 Работа с модулями: создание, подключение инструкциями import и from

27.1 Подключение модуля из стандартной библиотеки.
27.2 Использование псевдонимов.
27.3 Инструкция from.
27.4 Создание своего модуля на Python.

28 Объектно-ориентированное программирование. Общее представление

29 Инкапсуляция, наследование, полиморфизм

29.1 Инкапсуляция.
29.2 Наследование.
29.3 Полиморфизм.

30 Перегрузка операторов

30.1 Перегрузка арифметических операторов.

31 Декораторы

31.1 Передача декоратором аргументов в функцию.
31.2 Декорирование методов.
31.3 Декораторы с аргументами.
31.4 Некоторые особенности работы с декораторами.
31.5 Примеры использования декораторов.

32 Устанавливаем python-пакеты с помощью pip

32.1 Установка pip.
32.2 Начало работы.
32.3 Что ещё умеет делать pip.

33 Часто задаваемые вопросы

33.1 Почему я получаю исключение UnboundLocalError, хотя переменная имеет значение?
33.2 Каковы правила для глобальных и локальных переменных в Python?

Тематический план

№	Название темы	Количество часов
1	***Возможности языка Python***	***1***
2	***Скачать Python***	***1***
3	***Первая программа. Среда разработки IDLE***	***1***
4	***Синтаксис языка Python***	***1***
5	***Программа не работает. Что делать?***	***1***
6	***Инструкция if-elif-else, проверка истинности, трехместное выражение if/else***	***2***
7	***Циклы for и while, операторы break и continue, волшебное слово else***	***3***
8	***Ключевые слова, модуль keyword***	***1***
9	***Встроенные функции***	***1***
10	***Числа: целые, вещественные, комплексные***	***2***
11	***Работа со строками в Python: литералы***	***1***
12	***Строки. Функции и методы строк***	***1***
13	***Форматирование строк. Метод format***	***1***
14	***Списки (list). Функции и методы списков***	***2***
15	***Индексы и срезы***	***2***
16	***Кортежи (tuple)***	***3***
17	***Словари (dict) и работа с ними. Методы словарей***	***1***
18	***Множества (set и frozenset)***	***1***
19	***Функции и их аргументы***	***2***
20	***Исключения в python. Конструкция try - except для обработки исключени***	***1***

21	***Байты (bytes и bytearray)***	***1***
22	***None (null), или немного о типе NoneType***	***2***
23	***Файлы. Работа с файлами.***	***2***
24	***With ... as - менеджеры контекста***	***1***
25	***PEP 8 - руководство по написанию кода на Python***	***6***
26	***Документирование кода в Python. PEP 257***	***2***
27	***Работа с модулями: создание, подключение инструкциями import и from***	***4***
28	***Объектно-ориентированное программирование. Общее представление***	***1***
29	***Инкапсуляция, наследование, полиморфизм***	***3***
30	***Перегрузка операторов***	***1***
31	***Декораторы***	***4***
32	***Устанавливаем python-пакеты с помощью pip***	***2***
33	***Часто задаваемые вопросы***	***3***
34	***Задачи по Python***	***8***
	Итого:	**69**