

*Кондраткова Татьяна Алексеевна,
учитель информатики,
педагог дополнительного образования,
г. Санкт-Петербург*

Образовательная программа дополнительного образования детей «Основы программирования на языке Python»

(для детей 13-17 лет, срок реализации 1 год, уровень освоения: общекультурный).

АННОТАЦИЯ

Программа «**Основы программирования на языке Python**» предназначена прежде всего для учащихся старших классов естественнонаучных лицеев, для тех, кто свою будущую профессию связывает с информационными технологиями и программированием.

Программа предназначена для всех, кто любит решать сложные и интересные задачи, для тех, кто хочет знать универсальные методы решения сложных задач с использованием современных компьютерных технологий.

Персональные компьютеры широко внедряются в науку и технику, управленческую деятельность, технологические процессы. Программирование применяется в различных областях науки и техники является одним из основных способов реализации технологических решений.

Отечественная промышленность испытывает огромную потребность в высококвалифицированных специалистах и, прежде всего, в специалистах занятых в наукоёмких областях, где применяются инновационные технологии.

Профессия программист не только увлекательна, но и одна из наиболее востребованных на рынке труда. Особенно востребованы специалисты по языку программирования Python.

Python – это простой в освоении и мощный язык программирования. Он предоставляет эффективные высокоуровневые структуры данных, а также простой, но эффективный подход к объектно-ориентированному программированию. Его синтаксис и динамическая типизация наряду с тем, что он является интерпретируемым, делают его идеальным языком для написания сценариев и быстрой разработки приложений в различных областях и на большинстве платформ.

Для успешной работы программист должен знать определенный минимальный набор алгоритмов, а также владеть способами их программной реализации.

Программа обучения предусматривает изучение основных возможностей среды разработки Python и методов работы в ней, изучение фундаментальных понятий программирования и базовых структур данных.

Для освоения программы предварительного знания языков программирования и практики программирования на каком-либо другом языке не требуется.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Образовательная программа «**Основы программирования на языке Python**» предназначена для реализации в системе дополнительного образования во внеурочное время.

Направленность программы – техническая.

Основы инженерного и технического образования – а именно такие специалисты сегодня, да и в ближайшем будущем будут остро нужны стране – закладываются именно в школе.

Программа «Основы программирования на языке Python» соответствует Концепции инженерно-технической подготовки школьников.

Концепция инженерно-технической подготовки школьников разработана на основании поручения Президента Российской Федерации от 4 мая 2016г с учетом Стратегии научно-технологического развития Российской Федерации, утвержденной Указом Президента Российской Федерации от 1 декабря 2016г. №642, Национальной технологической инициативы, (постановление Правительства Российской Федерации от 18 апреля 2016г. №317 «О реализации Национальной технологической инициативы») и Программы «Цифровая экономика Российской Федерации», утвержденной распоряжением Правительства Российской Федерации от 28 июля 2017г. №1632-р.

Среди задач, сформулированных в Концепции:

- изучение элементов как традиционных, так и наиболее перспективных технологических направлений;
- формирование у обучающихся культуры проектной и исследовательской деятельности;
- формирование ключевых навыков в сфере информационных и коммуникационных технологий ;
- создание системы выявления, оценивания и продвижения обучающихся, обладающих высокой мотивацией и способностями в сфере материального и социального конструирования, включая инженерно-технологическое направление и ИКТ.

Уровень освоения программы – общекультурный.

Актуальность и новизна образовательной программы

Программа представляет особый интерес в связи с широким распространением цифровой техники в обществе, в связи с возрастающей потребностью обучающихся в освоении цифровых технологий и повышением их интереса к современным языкам программирования.

В век информационных технологий практически каждый школьник к 7-8 классу уже владеет навыками работы на персональном компьютере и работы в сети Интернет.

Содержание программы предусматривает ознакомление воспитанников с основами работы в интегрированной среде программирования и основными возможностями языка программирования Python. Программа позволяет освоить практически все операторные конструкции и познакомиться с основными функционального и объектного программирования.

Воспитанники, прошедшие обучение по данной программе получают знания, умения и навыки, необходимые для сознательного выбора в будущем профессии связанной с программированием. Программа позволяет школьникам познакомиться и понять тонкости профессии программист и профессий связанных с разработкой IT-приложений, оценить себя в этих профессиях, выработать профессионально важные качества.

Отличительной особенностью программы является её направленность на формировании у воспитанников потребности в инженерном образовании и научном исследовании.

Инновационной является авторская методика, позволяющая сочетать изучение достаточно сложного языка программирования с исследовательской деятельностью и разработкой проектов самых разнообразных по содержанию, позволяющая ребенку в конце курса обучения почувствовать себя настоящим профессионалом.

Цель образовательной программы

Создание условий для достижения обучающимися результатов развития в личностном, предметном, метапредметном направлениях, обеспечивающих их социальную адаптацию в области программирования, ИКТ на профессиональном уровне.

Развить личность обучающегося, способного к творческому самовыражению через овладение основами программирования на одном из самых востребованных языков программирования Python.

Понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации; научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи образовательной программы

Образовательные:

- Познакомить с возможностями и особенностями современного языка программирования Python;
- 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.
- Сформировать систему знаний, умений и навыков, необходимых в работе программиста;
- Обучить технологии работы со справочниками по языку программирования и поиску нужной информации на сайтах разработчиков программного обеспечения;
- Обучить приемам предъявления результатов проделанной работы на конференциях и конкурсах;
- Обучить коммуникативным навыкам при совместной работе над проектом.

Развивающие:

- Развить познавательные потребности и способности школьников
- Развивать творческие способности, алгоритмическое и логическое мышление;
- Развивать представления о возможностях применения современных компьютерных технологий в профессиональной деятельности;
- Развивать внимательность и наблюдательность, прививать навыки аккуратности и точности в работе;
- Повысить интеллектуальный уровень и расширить интеллектуальные навыки;
- Создать мотивацию к постоянному самообразованию.

Воспитательные:

- Воспитывать навыки взаимодействия при командной работе над проектом;
- Воспитывать чувство ответственности за результаты деятельности;
- Способствовать формированию культуры программирования;
- Способствовать формированию у воспитанников культуры создания и этики представления проектов на конференциях и в Интернете;
- Сформировать осознанное отношение к выбору будущей профессии.

Предметные: изучить принципы структурного, функционального и объектного программирования на примере языка программирования Python. Научиться формулировать и анализировать алгоритмы, составлять и отлаживать программы.

Метапредметные: научиться моделировать различные процессы реального мира в компьютере и использовать такое моделирование как инструмент познания.

Личностные: гармоничное развитие творческих способностей и логического мышления учащихся.

Условия реализации программы

Программа дополнительного образования ориентирована на подростков в возрасте от 13 до 17 лет.

Набор воспитанников осуществляется на добровольной основе, по интересам. Ограничения могут быть только по медицинским показаниям. Состав групп разновозрастной, в зависимости от базовой подготовки подростка. При условии наличия определённых навыков обучающийся может быть зачислен на другой модуль или получать дополнительные консультации, включая дистанционные, по сложным темам за рамками программы.

Срок реализации программы: 1 год, 68 часов в год. Итого: 68 часов.

Форма обучения – очная, в группах по 12 обучающихся.

Режим занятий: 1 раз в неделю по 2 часа.

Соотношение теории и практики

Занятия проводятся в компьютерном классе оснащённом индивидуальными стационарными компьютерами и оборудованием для презентаций.

В каждом занятии прослеживаются три части: теоретическая, практическая и исследовательская.

Учебные часы условно делятся на теорию и практику с соотношением 30 часов теория и 34 часа практика. Однако, теория включает в себя не только разбор конструкции языка программирования, алгоритмических приёмов, технологии решения задач, но и практические задания, выполнение которых непосредственно не связано с работой на персональном компьютере. На теории организуется обсуждение с учащимися сложных вопросов, проектных решений, применение тех или иных методов в различных профессиях.

Практические занятия проводятся непосредственно на персональном компьютере, и предполагают работу в интегрированной среде программирования Python с 10 минутным перерывом, во время которого учащиеся выполняют упражнения для снятия зрительного напряжения и снятия напряжения с костно-мышечного аппарата.

Исследовательская часть предполагает всестороннее изучение возможностей отдельных конструкций языка программирования Python и возможность их применения при решении нестандартных задач прикладного характера, разработку оптимальных нестандартных алгоритмов, создание компактного программного кода. Обучающиеся могут предложить собственный алгоритм решения задачи и протестировать работу соответствующей программы.

Формы организации деятельности воспитанников на занятии:

- фронтальная работа;
- индивидуальная работа;
- работа в малых группах;
- коллективная работа;
- работа с электронным учебником.

Формы проведения занятий:

- учебное занятие;
- лекция;
- мастер-класс;
- мастерская;
- творческая встреча с профессионалами;
- защита проектов.

Педагогическая целесообразность.

Программа направлена на социализацию современных школьников путём знакомства их с профессией программист и профессиональным программированием. Одной из особенностей

программы является развитие у обучающихся алгоритмического и логического мышления. В процессе освоения программы у воспитанников формируются организационные навыки и профессиональные качества важные не только в профессии программист, но и необходимые для любого члена общества.

Методы обучения

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 педагогом, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе — обучение учащихся навыкам самостоятельной творческой деятельности.

Методы проведения занятий

1. Словесные методы обучения:

- устное изложение материала с презентационным сопровождением;
- беседа,

2. Наглядные методы:

- показ программного кода и результатов его выполнения на тестовых данных;
- показ приёмов отладки программного кода педагогом;
- показ приёмов работы со средой разработки;
- демонстрации различных вариантов оптимизации программного кода.

3. Практические методы обучения:

- Написание и отладка программ.

Методы в основе которых лежит уровень деятельности детей:

- объяснительно-иллюстративные методы обучения;
- репродуктивные методы обучения;
- частично-поисковые методы обучения;
- проектный метод обучения.

Программа не создает учебных перегрузок для школьников: содержание построено таким образом, что изучение всех последующих тем обеспечивается предыдущими, материал дробится на простые легко воспринимаемые части и полностью усваивается на занятии, домашнее задание носит творческий характер и выполняется по желанию учащихся.

Санитарно-гигиенические требования при работе в компьютерном классе обеспечены.

Программа состоит из отдельных тем, которые должны изучаться последовательно. В зависимости от предварительной подготовки обучающихся в группе количество часов на изучение темы может варьироваться.

Ожидаемые результаты

В результате обучения по программе «Основы программирования на языке Python» обучающиеся овладеют основными навыками программирования на языке Python и будут готовы к дальнейшему самостоятельному изучению языка программирования и совершенствованию своих навыков в области программирования. Обучающиеся будут иметь представление о профессии программист и о своих возможностях преуспеть в данной профессии.

По окончании курса обучения учащиеся:

Познакомятся с технологией создания компьютерных программ, и современными концепциями программирования;

Узнают особенности и возможности языка программирования Python и области его применения;

Научатся составлять алгоритмы для решения задач и реализовывать эти алгоритмы на компьютере в виде программ, написанных на языке Python;

Научатся работать в интерактивной среде программирования и использовать её основные возможности для отладки и тестирования программ;

Получат общие представления о характере труда программиста и специфике его работы;

Овладеют основами функционального программирования на языке Python;

Приобретут навыки объектного программирования;

Смогут самостоятельно создавать и отлаживать несложные приложения на языке программирования Python;

Освоят нестандартные приёмы программирования;

Результатом обучения будет

- портфолио электронных учебных работ обучающегося;
- самостоятельно выполненный проект.

Виды и формы контроля:

Текущий контроль знаний проводится на каждом занятии как

- текущий рефлексивный самоанализ;
- контроль и самооценка учащимися выполняемых заданий;
- в форме взаимопроверки;
- в форме фронтального опроса;
- в форме письменного опроса;
- в форме выполнения практического задания.

Тематический контроль осуществляется по завершении крупного блока (темы). Он позволяет оценить знания и умения учащихся, в ходе достаточно продолжительного периода работы. Тематический контроль проводится

- в форме собеседования;
- в форме предъявления программного продукта;
- в форме практической работы ориентированной на применение полученных знаний в нестандартной ситуации.

Итоговый контроль проводится по завершении учебного курса в форме защиты портфолио.

Формы подведения итогов реализации дополнительной образовательной программы

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Практическим результатом обучения воспитанников является также самостоятельно выполненный проект, включающий в себя не только компьютерную программу, но и описание алгоритма её работы, контрольные тесты для её проверки, описание используемых языковых конструкций и оригинальных приёмов программирования, область практического применения и другую документацию, отражающую работу над созданием проекта.

Темы проектов выбираются учащимися как самостоятельно, так и из числа предложенных преподавателем, но всегда обсуждается в группе. Необходимо чтобы программа имела практическое применение и была интересна не только авторам проекта. Воспитанник должен почувствовать важность выбранной им темы в глазах общества, желание показать своё мастерство и уникальность технологического решения созданной им программы.

Обеспечивается возможность презентации проекта на школьной научно-практической конференции, конкурсе или фестивале проектов.

Примерные требования при защите проекта

Учащиеся должны (по желанию) продемонстрировать самостоятельно разработанную программу, показать владение техническими, программными средствами разработки проекта, знание основ программирования, уметь ответить на вопросы по содержанию.

По техническим средствам:

- какие технические средства были задействованы при создании проекта;
- объяснить минимальные требования к программному и аппаратному обеспечению для работы программы.

По технологии разработки проект:

- охарактеризовать программные средства, используемые при выполнении проекта;
- объяснить технологические приемы разработки;
- наличие документации;
- рекомендации по работе с программой.

По разработке программы:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- код должен быть «читаемым» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.);
- программа должна выполнять поставленные задачи, и правильно работать;
- наличие собственных конструктивных решений;
- знание основ программирования и знание технология программирования;
- наличие дружественного интерфейса.

По содержанию проекта:

- актуальность работы,
- глубина и логика раскрытия темы,
- форма подачи информации,
- заинтересованность в проведении исследования,
- области применения разработки,
- источники информации.

Оценка образовательных результатов освоения программы

При подведении итогов учитывается не только результаты диагностики и качество выполнения учебных заданий, но и личные достижения учащихся. Рекомендуется в начале и конце курса обучения заполнять информационную карту:

«Информационная карта освоения образовательной программы»

№ п/п	ФИО учащегося	Месяц						Общий суммарный балл	Уровень освоения
		Критерии наблюдения							
		А	Б	В	Г	Д	Е		

Критерии:

«А» Социальная активность обучающегося;

«Б» Коммуникабельность, умение работать в коллективе;

«В» Владение навыками алгоритмизации и знание основных конструкций языка программирования;

«Г» Навыки анализа, синтеза и культуры представления информации;

«Д» Умение работать на персональном компьютере в среде программирования;

«Е» Владение технологией программирования;

«Ж» Уровень самостоятельности;

По каждому критерию выставляется: от 0 до 5 баллов. Общий уровень определяется суммированием баллов:

- низкий уровень: от 7 до 20 баллов;
- средний уровень: от 21 до 27 баллов;
- высокий уровень: от 28 до 35 баллов.

При разработке данной программы использованы:

Рабочая программа дистанционного курса «Основы программирования на языке Python».

Автор Литвинов В.Н. Лицей №136. Новосибирск;

Программа курса «Информатика», углубленный уровень, К.Ю. Поляков М.: Бином.

Лаборатория Знаний, 2013 год;

Рабочая программа «Математическое инженерное и спортивное объединения школьников Как приручить питона?» (основы спортивного и прикладного программирования для школьников)

Филиппов В.А. ООО ЦДО «АЙТИЛЁН», г. Псков.

Тематический план

№ п/п	Название темы	Количество часов
I	Основы охраны труда. Организация рабочего места.	2
II	Знакомство с языком Python. Установка программы Python.	4
III	Переменные и выражения. Организация ввода и вывода данных. Операции.	6
IV	Условные предложения. Программирование ветвящихся алгоритмов.	6
V	Циклы	8
VI	Функции	8
VII	Модули	4
VIII	Структуры данных	14
IX	Стиль программирования и отладка программ	2
XI	Работа над проектами	10
Всего:		68 часов

Учебно-тематический план

№ п/п	Название темы	Количество часов		
		Всего	Теория	Практика
I.	Основы охраны труда. Организация рабочего места.	2	1	1
II.	Знакомство с языком Python. Установка программы Python.	4		
2.1	Общие сведения о языке Python. Установка программы Python.	2	1	1
2.2	Режимы работы. Знакомство с языком Python	2	1	1
III	Переменные и выражения. Операции. Организация ввода и вывода данных.	6		
3.1	Переменные	1	1	
3.2	Выражения.	1		1
3.3	Операции. Элементарные действия с числами	2	1	1
3.4	Ввод и вывод данных	2	1	1

IV.	Программирование ветвящихся алгоритмов.	6		
4.1	Логические выражения и операторы. Условный оператор IF. Множественное ветвление	2	1	1
4.2	Реализация ветвления в языке Python.	2	1	1
4.3	Решение задач на программирование ветвящихся алгоритмов.	2		2
V	Циклы. Программирование циклических алгоритмов.	8		
5.1	Оператор цикла с условием Оператор while	2	1	1
5.2	Оператор цикла с параметром FOR	2	1	1
5.3	Вложенные циклы	2	1	1
5.4	Реализация циклических алгоритмов	2		2
VI	Функции	8		
6.1	Создание функций	2	1	1
6.2	Локальные и глобальные переменные	2	1	1
6.3	Решение задач с использованием функций	2		2
6.4	Рекурсивные функции	2	1	1
VII	Модули	4		
7.1	Модули в языке Python	2	1	1
7.2	Оформление собственных модулей.	2	1	1
VIII	Структуры данных	14		
8.1	Строки. Срезы строк.	2	1	1
8.2	Списки. Срезы списков.	2	2	
8.3	Кортежи	2	1	1
8.4	Словари	2	1	1
8.5	Последовательности	2	1	1
8.6	Матрицы. Множества	2	1	1
8.7	Ссылки	2	1	1
IX	Стиль программирования и отладка программ	2		
9.1	Стиль программирования	1	1	
9.9	Отладка программ	1	1	
X	Работа над проектами	10		
10.1	Требования к проектам. Выбор темы.	2	1	1
10.2	Разработка алгоритма и Написание программного кода.	2		2
10.3	Работа над тестированием и отладкой программ	2		2

10.4	Работа над тестированием и отладкой программ	2		2
10.5	Оформление проектной документации	1		1
10.6	Защита проектов	1		1
				Итого: 68 часов

Содержание программы

I. Основы охраны труда. Организация рабочего места. Валеологические аспекты и требования безопасности при работе за компьютером. Безопасность в Интернете.

Обучающиеся знакомятся с правилами поведения в компьютерном классе, правилами охраны труда, рекомендациями по организации рабочего места.

Практическая работа 1.1. Организация рабочего места.

Обучающиеся должны уметь:

- правильно организовать рабочее место;
- следить за временными нормами работы за персональным компьютером;
- использовать компьютер, не нанося вреда своему здоровью;

Обучающиеся знакомятся с правилами безопасного поведения при работе в сети Интернет.

II. Знакомство с языком Python. Установка программы Python.

Общие сведения о языке Python, преимущества языка, сферы применения. Установка программы Python на компьютер. Режимы работы Python. Использование командной строки интерпретатора. Выбор редактора. Работа со средой программирования. Сохранение, открытие и запуск программ. Получение помощи. Знакомство с языком Python и написание простейших программ. Структура программы на языке Python Комментарии.

Практическая работа 2.1. Установка программы Python.

Практическая работа 2.2. Режимы работы с Python .

Обучающиеся должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Обучающиеся должны уметь:

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

III. Переменные и выражения. Операции. Организация ввода и вывода данных.

Литеральные константы, числа, строки. Использование кавычек. Объединение строковых констант. Метод format/

Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Объекты. Примеры использования переменных и констант. Логические и физические строки. Отступы. Операторы и их применение. Выражения. Операции. Порядок выполнения операций. Краткая запись математических операций и порядок вычисления. Изменение порядка вычисления. Математические функции. Композиция. Ассоциативность. Элементарные действия с числами.

Ввод и вывод данных. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных.

Практическая работа 3.1. Работа со справочной системой .

Практическая работа 3.2. Переменные.

Практическая работа 3.3. Выражения.

Практическая работа 3.5. Задачи на элементарные действия с числами.

Обучающиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания.

IV. Программирование ветвящихся алгоритмов.

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Поток команд. Ветвящиеся алгоритмы. Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python. Решение задач на программирование ветвящихся алгоритмов.

Практическая работа 4.1. Логические выражения.

Практическая работа 4.2. "Условный оператор" .

Практическая работа 4.3. Множественное ветвление.

Практическая работа 4.4. "Условные операторы".

Обучающиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not.

Обучающиеся должны уметь:

- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

V. Циклы. Программирование циклических алгоритмов.

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. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Реализация циклических алгоритмов. Решение задач с использованием циклов. Операторы управления циклом. Оператор break. Оператор continue.

Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 5.1. Решение задачи с циклом while.

Практическая работа 5.2. Решение задачи с циклом for.

Практическая работа 5.3. Реализация циклических алгоритмов.

Практическая работа 5.4. Случайные числа.

Практическая работа 5.5. Решение задач с циклом.

Обучающиеся должны знать / понимать:

- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Обучающиеся 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.

VI. Функции

Создание функций. Параметры функций. Локальные и глобальные переменные. Зарезервированные слова «global», «nonlocal». Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Значения аргументов по умолчанию. Ключевые аргументы. Переменное число параметров. Ключевые параметры. Оператор «return».

Строки документации. Аннотации. Решение задач с использованием функций. Рекурсивные функции.

Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 6.1. Создание функций.

Практическая работа 6.2. Локальные переменные.

Практическая работа 6.3. Решение задач с использованием функций.

Практическая работа 6.4. Рекурсивные функции.

Обучающиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Обучающиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

VII Модули.

Для чего нужны модули. Оформление модулей. Импорт модуля. Оператор `from ... import`. Имя модуля, `name`. Создание собственных модулей. Функция `dir`. Пакеты.

Практическая работа 7.1. Использование модулей.

Практическая работа 7.2. Создание модулей.

Обучающиеся должны знать / понимать:

- для чего нужны модули.

Обучающиеся должны уметь:

- использовать готовые модули;
- создавать собственные модули.

VIII. Структуры данных

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.

Списки. Объекты и классы. 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.

Последовательности.

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 8.1. Строки

Практическая работа 8.2. Решение задач со строками.

Практическая работа 8.3. Списки.

Практическая работа 8.4. Решение задач со списками.

Обучающиеся должны знать / понимать:

- сложные типы данных;
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;

- операции со строками;
- 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.

Обучающиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- 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.

IX. Стиль программирования и отладка программ.

Процесс разработки программного обеспечения. Стиль программирования. Отладка программ.

Обучающиеся должны знать / понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Обучающиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

X. Работа над проектами

Требования к проектам. Выбор темы. Разработка алгоритма и написание программного кода. Работа над тестированием и отладкой программ. Оформление проектной документации. Защита проектов.

Методическое обеспечение образовательной программы

1. Методическое пособие для учителя к завершённой предметной линии учебников «Информатика» для 10–11 классов общеобразовательных учреждений (углублённый уровень) Автор-составитель: К.Ю. Поляков, Е. А. Еремин. БИНОМ. Лаборатория знаний / Автор-составитель: М. Н. Бородин.—Эл. изд.—М. : БИНОМ. Лаборатория знаний, 2013 ISBN 978-5-9963-1344-

2. «Здоровье и безопасность детей в мире компьютерных технологий и Интернета» Электронное методическое пособие. Партнёрство в образовании. Проект «Академия учителей»

Материально-технические и информационно-технические средства обучения

Технические средства:

1. Персональные компьютеры – 12 рабочих станций;
2. Рабочее место учителя – персональный компьютер, принтер, сканер, звуковые колонки.
3. Оборудование для презентаций – мультимедийный проектор, экран;
4. Маркерная доска.

Программное обеспечение:

1. Операционная система Windows или Linux;
2. Интегрированные среды программирования Python 3.3 - Python 3.5 и выше.

Цифровые образовательные ресурсы:

1. Л.Н. Горбунов и др. УМК Здоровье и безопасность детей в мире компьютерных технологий и Интернет. Электронное приложение. Москва. САЛОН-ПРЕСС. 2010;
2. Федеральный портал Единая коллекция образовательных ресурсов (school-collection.edu.ru/)
3. Домашняя страница Python www.python.org . Справочные материалы, официальная документация;
4. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>;
5. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>;
6. Сайт проекта Open Book Project openbookproject.net содержит серию практических примеров на Python Криса Мейерса;
7. Интерактивный Python. Trinket <https://trinket.io/python>;
8. Интерактивный учебник языка Питон <http://pythontutor.ru/> ;
9. Python 3 для начинающих <https://pythonworld.ru/> ;
10. Python GUI Programming (Tkinter) https://www.tutorialspoint.com/python/python_gui_programming.htm <http://www.tkdocs.com/tutorial/> и <http://effbot.org/tkinterbook/> ;
11. Python IDE for beginners <http://thonny.org/> ;
12. Бесплатный, простой и профессиональный инструмент для изучения программирования с помощью Python <https://www.jetbrains.com/pycharm-edu/> ;
13. 5. Язык Python (Электронный ресурс) <http://kpolyakov.spb.ru/school/probook/python.htm>.

Дидактические материалы:

1. Раздаточный материал контролирующего и обучающего характера по каждой теме.
2. Задания на практические работы.
3. Карточки с индивидуальными заданиями.
4. Раздаточный материал справочного характера.
5. Раздаточный материал теоретического характера.
6. Демонстрационные материалы в электронном виде.

Список литературы

Литература для педагога

1. Поляков К.Ю., Еремин Е.А. Информатика. 10-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013.
2. Поляков К.Ю., Еремин Е.А. Информатика. 11-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013.
3. Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
4. Python. Справочник Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.

Литература для учащихся

1. К.Ю. Поляков, Е. А. Еремин. «Информатика», углубленный уровень, М.: Бином. Лаборатория Знаний, 2013.
2. М. Лутц «Изучаем Питон», Санкт-Петербург: Символ, 2011г.
3. Самоучитель Python. Дмитрий Мусин.2016 pythonworld.ru