

Комитет по делам образования города Челябинска

Муниципальное Бюджетное Общеобразовательное Учреждение
«Средняя общеобразовательная школа № 131 г. Челябинск»

Согласовано

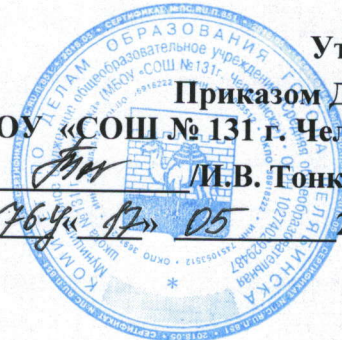
Протокол педагогического совета №3

«15» 05 2023 г.

Утверждено

Приказом Директора
МБОУ «СОШ № 131 г. Челябинска»

И.В. Тонконоженко
№ 76-г/17 05 2023 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«Программируем на Python»**

Срок реализации программы: 1 год

Возраст детей: 14-16 лет

Разработчик:

Юсупова Ирина Анатольевна,
учитель информатики

Челябинск
2023г.

Содержание

1.	«Комплекс основных характеристик программы»	
1.1	Пояснительная записка	3-4
1.2	Цель и задачи программы	5
1.3	Содержание программы (учебный план и содержание учебно-тематического плана)	6-11
1.4	Планируемые результаты	12-13
2.	«Комплекс организационно-педагогических условий»	
2.1	Календарный учебный график	14
2.2	Условия реализации программы	14
2.3	Формы аттестации	14
2.4	Оценочные материалы	15
2.5	Методические материалы	16
	Список литературы	17
	Приложения	18-20

Раздел 1. «Комплекс основных характеристик программы»

1.1 Пояснительная записка

Данная программа является дополнительной общеобразовательной общеразвивающей технической направленности, очной формы обучения, сроком реализации 1 год, для детей 14-16 лет, уровень освоения – основное общее образование.

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Программа «Программирование на Python» имеет **техническую направленность**, в её основу заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на детальное изучение алгоритмизации, реализацию межпредметных связей, организацию проектной и исследовательской деятельности обучающихся.

Проект программы составлен в соответствии с государственными требованиями к образовательным программам системы дополнительного образования детей на основе следующих нормативных документов:

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
2. Приказ Министерства просвещения Российской Федерации от 09 ноября 2018 года № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
3. Письмо Минобрнауки РФ от 18.11.2015 № 09-3242 «О направлении рекомендаций» (вместе «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»;
4. Распоряжение Правительства Российской Федерации от 4 сентября 2014 года № 1726-р «Концепция развития дополнительного образования детей»;
5. Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р
6. «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;

7. Постановление Правительства Российской Федерации от 30 декабря 2015 года № 1493 «О государственной программе «Патриотическое воспитание граждан Российской Федерации на 2016-2020 годы»;

8. Постановление Главного государственного санитарного врача РФ от 04.07.2014 № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей»;

9. Федеральная целевая программа развития образования на 2016-2020 годы, утвержденная Постановлением Правительства Российской Федерации от 23 мая 2015 года № 1499;

10. Устав МБОУ «СОШ №131 г. Челябинска»;

11. Положение об организации и осуществлении образовательной организацией деятельности по дополнительным общеобразовательным общеразвивающим программам в Муниципальном бюджетном общеобразовательном учреждении «Средняя общеобразовательная школа №131 г. Челябинска».

Актуальность данной программы вызвана потребностью современного информационного общества в высокообразованных, адаптированных к изменениям специалистах в IT-сфере. Для удовлетворения данной потребности перед дополнительным образованием стоит задача развития человеческого потенциала через выявление талантливых детей, развитие их мотивации и способностей.

Программа разработана с учетом возрастных особенностей детей.

Новизна дополнительной образовательной общеразвивающей программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы также разработаны с помощью языка программирования Python.

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

Отличительная особенность данной программы заключается в том, что основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

Особенности возрастной группы. Возраст обучающихся, участвующих в реализации данной образовательной программы 14-16 лет. Обучающиеся этого возраста способны на высоком уровне усваивать разнообразную информацию о программировании.

Уровень освоения программы. Программа имеет базовый уровень сложности.

Объем программы – 72 часа.

Срок освоения программы – 1 год.

Режим занятий: 1 раз в неделю по 2 часа.

Условия реализации программы. На обучение по Программе принимаются все желающие, не имеющие противопоказаний по состоянию здоровья. Набор учащихся осуществляется на добровольной основе, без специального отбора. Количественный состав детей в группе до 15 человек. Состав группы постоянный. Зачисление в группы производится на основании заполнения родителями заявления.

1.2. Цель и задачи программы

Цель программы: Формирование интереса к техническим видам творчества, развитие логического, технического мышления, создание условий для творческой самореализации личности ребёнка посредством получения навыков разработки эффективных алгоритмов, для реализации их в виде программы, написанной на языке программирования Python.

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных задач:

Задачи:

Образовательные:

- изучение конструкций языка программирования Python;
- знакомство с принципами и методами функционального и объектно-ориентированного программирования; основными структурами данных и типовыми методами обработки этих структур;
- формирование навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

Развивающие:

- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развитие у обучающихся интереса к программированию, самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- формирование и развитие навыков работы с различными источниками информации, необходимой для решения учебных задач; умения планировать свои действия с учётом фактора времени, в обстановке с элементами конкуренции, предвидеть результат и достигать его, при необходимости вносить коррективы в первоначальный замысел.

Воспитательные:

- воспитание этики групповой работы;
 - воспитание отношений делового сотрудничества, взаимоуважения;
 - развитие основ коммуникативных отношений внутри проектных групп и в коллективе в целом;
 - воспитание упорства в достижении результата
 - формирование активной жизненной позиции, гражданско-патриотической ответственности;
- формирование целеустремлённости, организованности, неравнодушия, ответственного отношения к труду, толерантности и уважительного отношения к окружающим.

1.3. Содержание программы (учебный план и содержание учебного плана)

Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в Python	7	3	4	
1.1.	Язык Python. Среда программирования. Особенности ввода-вывода	2	1	1	Практическое задание
1.2.	Типы данных, операции. Оператор присваивания	2	1	1	Выполнение упражнений
1.3.	Числа. Стандартные операции	3	1	2	Практическое задание
2.	Алгоритмические инструкции	11	3	8	
2.1.	Условный оператор	1	1	-	Опрос
2.2.	Цикл while. Теория чисел	5	1	4	Практическое задание
2.3.	Цикл for	5	1	4	Практическое задание
3.	Строки	10	4	6	
3.1.	Литералы строк	2	1	1	Практическое задание
3.2.	Срезы строк	2	1	1	Практическое задание
3.3.	Методы строк	6	2	4	Практическое задание
4.	Функции	10	4	6	
4.1.	Парадигма и преимущества структурного программирования	1	1	-	Опрос
4.2.	Граф и стек вызовов функций. Области видимости переменных	2	1	1	Практическое задание

4.3.	Прямая рекурсия	5	1	4	Практическое задание
4.4.	Косвенная рекурсия	2	1	1	Решение задач повышенной трудности
5.	Списки и кортежи	10	4	6	
5.1.	Списки и кортежи в Python. Сходства и различия	1	1	-	Опрос
5.2.	Операции со списками	5	1	4	Практическое задание
5.3.	Срезы списков	2	1	1	Практическое задание
5.4.	Матрицы. Операции над матрицами	2	1	1	Решение задач повышенной трудности
6.	Словари и множества	4	2	2	
6.1.	Словари	2	1	1	Практическое задание
6.2.	Множества	2	1	1	Практическое задание
7.	Объектно-ориентированное программирование	19	2	17	
7.1.	Классы в Python	1	1	-	Опрос
7.2.	Разработка собственного класса	3	1	2	Практическое задание
7.3.	Разработка и программирование собственного проекта	15	-	15	
8.	Заключительное занятие	1	-	1	
8.1.	Подведение итогов. Индивидуальный проект.	1		1	Защита проекта
	Всего	72	22	50	

Содержание программы

1. Введение в Python

Основные понятия: трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

1.1. Язык Python. Среда программирования. Особенности ввода- вывода

Теория. Язык программирования Python. Достоинства и недостатки.
Области применения. Интерактивный режим работы программы.

Практика. Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

1.2. Типы данных, операции. Оператор присваивания

Теория. Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

1.3. Числа. Стандартные операции

Теория. Стандартные операции с целыми и действительными числами.
Стиль программирования Python (PEP 8).

Практика. Решение простых задач в интерактивном режиме.

2. Алгоритмические инструкции

Основные понятия: логический тип данных, логические операции (and, or, not), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

2.1. Условный оператор

Теория. Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

2.2. Цикл while

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики.

2.3. Цикл for

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

3. Строки

Основные понятия: символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape- последовательности,

положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

3.1. Литералы строк

Теория. Понятие «литералы строк».

Практика. Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

3.2. Срезы строк

Теория. Форматирование строки.

Практика. Преобразование строки. Решение упражнений.

3.3. Методы строк

Теория. Методы работы со строкой.

Практика. Применение методов строки. Решение задач.

4. Функции

Основные понятия: подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции. Парадигма и преимущества структурного программирования

Теория. Обзор парадигм программирования. Особенности применения языков программирования.

4.1. Граф и стек вызовов функций. Области видимости переменных

Теория. Применение стек и граф вызовов функций.

Практика. Выполнение тренировочных упражнений.

4.2. Прямая рекурсия

Теория. Понятие прямой рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

4.3. Косвенная рекурсия

Теория. Понятие косвенной рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

Практические занятия к темам 4.2., 4.3., 4.4.: Нахождение суммы чисел. Числа Фибоначчи. Вычисление степени. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности.

5. Списки и кортежи

Основные понятия: список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

5.1. Списки и кортежи в Python. Сходства и различия

Теория. Представление списка и кортежа в памяти компьютера, сходства и различия.

5.2. Операции со списками

Теория. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка.

5.3. Срезы списков

Теория. Методы работы со списком и кортежем. Методы сортировки списка.

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

5.4. Матрицы. Операции над матрицами

Теория. Вычисление сложности алгоритма. Многомерные списки.

Практика. Решение задач повышенной трудности.

6. Словари и множества

Основные понятия: *словарь, множество, ключ, кодирование.*

6.1. Словари

Теория. Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

Практика. Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

6.2. Множества

Теория. Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

Практика. Решение задач повышенной трудности.

7. Объектно-ориентированное программирование (ООП)

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

7.1. Классы в Python

Теория. Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

7.2. Разработка собственного класса

Теория. Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

Практика. Создание собственного класса.

7.3. Разработка и программирование собственного проекта

Практика. Выбор вида и темы проекта. Составление технического задания.
Программирование. Разработка технической документации и презентации проекта.

7.4. Подведение итогов

Защита индивидуального проекта

1.5. Планируемые результаты

К концу обучения по программе будут достигнуты следующие результаты:

Предметные результаты:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- понимание основных предметных понятий («информация», «алгоритм», «исполнитель», «модель») и их свойств;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами – линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов;
- интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- умение соблюдать нормы информационной этики и права.

Личностные результаты:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;

- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Раздел 2. «Комплекс организационно-педагогических условий»

2.1. Календарный учебный график

Начало учебного года – 1 сентября.

Окончание учебного года – 31 мая.

Год обучения	Объем учебных часов	Всего учебных недель	Режим работы	Количество учебных дней
1 год обучения «Базовый»	72	36	1 раз в неделю по 2 часа	36

2.2. Условия реализации программы

Материально-техническое обеспечение:

Дидактический материал, необходимый для проведения занятий:

- краткие конспекты материалов для лекций;
- распечатки заданий для практикумов;
- презентационные материалы для объяснения;
- карточки с индивидуальными заданиями.

Техническое оснащение занятий:

- компьютер для демонстрации презентаций;
- проектор;
- рабочие компьютеры учащихся для работы с доступом в Интернет;
- принтер для распечатки заданий.

Кадровое обеспечение

Кадровое обеспечение разработки и реализации дополнительной общеобразовательной общеразвивающей программы осуществляется педагогом дополнительного образования, что закрепляется Профессиональным стандартом «Педагог дополнительного образования детей и взрослых (Приказ Минтруда России от 08.09.2015 г. № 613н).

2.3. Формы аттестации

В процессе реализации программы предусмотрены следующие виды контроля:

- **входной контроль** проводится с целью определения уровня знаний учащихся (Приложение № 1);
- **промежуточный контроль** проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;
- **итоговый контроль** – защита проекта.

Формы подведения итогов реализации программы

Результаты обучения по программе выявляются по итогам защиты проекта.

2.4. Оценочные материалы

Промежуточный контроль осуществляется путём наблюдения, определения качества выполнения заданий (Приложение 3), отслеживания динамики развития обучающегося (Приложение 2). Способы проверки уровня освоения тем: опрос, тестирование, решение задач, наблюдение, оценка выполненных самостоятельных работ.

Итоговая аттестация учащихся осуществляется по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

Баллы, набранные учащимся	Уровень освоения
0–50 баллов	Низкий
50–75 баллов	Средний
75–100 баллов	Высокий

2.5. Методические материалы

Особенности организации образовательного процесса: очная форма обучения.

Формы занятий

Форма проведения занятий очная. Занятия включают в себя теоретическую часть, с использованием репродуктивных приемов обучения и практическую деятельность - решения задач, за счет изучения материала модуля и работы с компьютерными программами.

При проведении занятий используются следующие формы работы:

- проблемно — поисковая, когда преподаватель ставит исследовательскую задачу перед учениками, и те должны, совместно с учителем найти наиболее подходящий способ решения;
- решение ситуационных производственных задач. Этот метод используется для формирования у учащихся профессиональных умений. Основным дидактическим материалом служит ситуационная задача, которая включает в себя условия (описание ситуации и исходные количественные данные) и вопрос (задание), поставленный перед учащимися. Ситуационная задача должна содержать все необходимые данные для ее решения
- демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном;
- самостоятельная, когда обучающиеся выполняют задание в течение занятия или нескольких занятий

Список литературы

Литература для обучающихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.
4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
5. Доусен М. Програмируем на Python / М. Доусен - СПб.: Питер, 2016. - 416с.
6. Лутц М. Изучаем Python, 4 издание / М. Лутц - СПб.: Символ-Плюс, 2011. - 1280 с.
7. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016. - 480с.
8. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений / Н.А. Прохоренок, В.А. Дронов - СПб.: «БХВ- Петербург», 2016. - 832с.
10. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А.Слинкин – М.: ДМК-Пресс, 2014. - 338с.

Цифровые образовательные ресурсы

1. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С., Задка М., Левис М.,Монтаро С., Реймонд Э.С., Кучлинг А.М., Лембург М.-А., Йи К.-П., Ксиллаг Д., Петрилли Х.Г., Варсав Б.А., Ахлстром Дж.К., Роскинд Дж., Шеменор Н.,Мулендер С. Язык программирования Python: [Электронный ресурс].URL: <https://goo.gl/8TzY8w>
2. Центр онлайн-обучения «Фоксфорд» <http://foxford.ru/>;
3. Национальный Открытый Университет «ИНТУИТ» <http://www.intuit.ru/>;
4. Образовательная платформа и конструктор онлайн-курсов Stepik. <https://stepik.org/course/67/syllabus>;
5. Всероссийский портал «Дистанционная подготовка по информатике» (informatics.mccme.ru);
6. Официальная документация языка Python (docs.python.org).
7. Codeacademy.com

Приложение № 1

1. Запишите значение переменной t , полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
<pre> алг нач цел t, it := 2 нц для i от 1 до 3t := t * i кц вывод t кон </pre>	<pre> DIM i, t AS INTEGER t = 2 FOR i = 1 TO 3 t = t * i NEXT i PRINT t </pre>	<pre> Vart, i: integer; Begin := 2; For i := 1 to 3 do t:=t * i; Writeln(t);End. </pre>

Ответ:

2. В таблице Dat хранятся данные о количестве краткосрочных командировок, в которые приходилось ездить сотруднику за последний год (Dat[1] — количество командировок в январе, Dat[2] — количество командировок в феврале и т. д.). Определите, что будет напечатано в результате выполнения следующего алгоритма, записанного на трёх алгоритмических языках.

Алгоритмический язык	Бейси к	Паскаль ь
<pre> алг нач целтаб Dat[1:12] цел k, m, month Dat[1] := 5 Dat[2] := 5 Dat[3] := 6 Dat[4] := 8 Dat[5] := 4 Dat[6] := 5 Dat[7] := 4 Dat[8] := 7 Dat[9] := 4 Dat[10] :=4 Dat[11] := 8 Dat[12] := 7 month:=1; m := Dat[1] нц для k от 2 до 12 если Dat[k] <= m то m:= Dat[k]; month := k все КЦ вывод month КОН </pre>	<pre> DIM Dat(12) AS INTEGER Dat[1] = 5 Dat[2] = 5 Dat[3] = 6 Dat[4] = 8 Dat[5] = 4 Dat[6] = 5 Dat[7] = 4 Dat[8] = 7 Dat[9] := 4 Dat[10] :=4 Dat[11] := 8 Dat[12] := 7 month = 1: m = Dat(1) FOR k = 2 TO 12 IF Dat(k) <= m THEN m = Dat(k) month = k END IF NEXT k PRINT month END </pre>	<pre> Var k, m, month: integer; Dat: array[1..12] of integer; Begin Dat[1] := 5; Dat[2] := 5; Dat[3] := 6; Dat[4] := 8; Dat[5] := 4; Dat[6] := 5; Dat[7] := 4; Dat[8] := 7; Dat[9] := 4; Dat[10] :=4; Dat[11] := 8; Dat[12] := 7; month:= 1; m := Dat[1]; for k := 2 to 12 do if Dat[k] <= m then begin m:= Dat[k]; month := k; end; write(month); End. </pre>

Ответ:

1. У исполнителя Квадратор две команды, которым присвоены номера: 1. возведи в квадрат, 2. прибавь 3.

Первая из них возводит число на экране во вторую степень, вторая — прибавляет к числу 3. Составьте алгоритм получения из числа 5 числа 127, содержащий не более 5 команд. В ответе запишите только номера команд. (Например, 12212 — это алгоритм: возведи в квадрат прибавь 3 прибавь 3 возведи в квадрат прибавь 3,

который преобразует число 2 в число 103).

Если таких алгоритмов более одного, то запишите любой из них. Ответ:

3. Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала вычисляется длина исходной цепочки символов; если она чётна, то в середину цепочки символов добавляется символ А, а если нечётна, то в начало цепочки добавляется символ Б. В полученной цепочке символов каждая буква заменяется буквой, следующей за ней в русском алфавите (А — на Б, Б — на В и т. д., а Я — на А). Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной была цепочка **ВРМ**, то результатом работы алгоритма будет цепочка **ВГСН**, а если исходной была цепочка **ПД**, то результатом работы алгоритма будет цепочка **РБЕ**.

Дана цепочка символов **ТОР**. Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (т. е. применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)? Русский алфавит: АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ.

Ответ:

4. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число — количество чисел, оканчивающихся на 6.

Пример работы программы:

Входные данные	Выходные данные
3, 16, 26, 24	о, z

Бланк наблюдения за динамикой личностного развития обучающихся

Направление / Группа _____

№ п/п	ФИО	ПОКАЗАТЕЛИ						
		Во время занятий проявляет устойчивый интерес и инициативу при освоении программы	Использует в общении базовую систему понятий	Активно сотрудничает со сверстниками, уважительно относится к мнению окружающих	Проявляет интерес к проектной деятельности, активно включается в групповую работу	Аккуратно относится к материально- техническим ценностям	Соблюдает правила безопасного поведения при работе с компьютерной техникой	Итого
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								

3 балла – качество проявляется систематически

2 балла – качество проявляется ситуативно

1 балл – качество не проявляется

Лист оценки работы обучающихся в процессе разработки

№ группы: _____

Дата: _____

№ п/ п	ФИ обучающегося	Сложность языка разработки (по шкале от 0 до 5 баллов)	Соответ- ствие написания программ- ного кода поставлен- ной задаче (по шкале от 0 до 5 баллов)	Презентация модели по плану. Степень владения специальным и терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригиналь- ности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруд- нений (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						