




МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «ЛИЦЕЙ №7»

Рассмотрена на заседании
педагогического совета

Протокол № 17 от 31.08.2022

Утверждаю
Директор

 С.В. Пилипушка
31.08 2022 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**
Технической направленности

«Программирование»

Возраст обучающихся: 13-15 лет

Срок реализации программы: 1 год, 72 часа

Авторы-составители программы:
Паклинов Артем Сергеевич,
учитель информатики

Пояснительная записка.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» составлена на основе курса К.Ю. Полякова «Программирование. Python, C++», М.:, БИНОМ. Лаборатория знаний, 2018 г., примерной программы внеурочной деятельности начального и основного образования, Федерального государственного образовательного стандарта основного общего образования, утвержден приказом Министерства образования и науки Российской Федерации от 17 декабря 2010 г. № 1897, Законом РФ «Об образовании» от 29.12.12 г. №273-ФЗ и письмом Минпросвещения России от 5.09.2018 г. №03-ПГ-МП-42216 «Об участии учеников муниципальных и государственных школ РФ во внеурочной деятельности».

Данная программа курса по предмету «Программирование на языке Python» основана на учебно-методическом комплекте (далее УМК), обеспечивающем обучение курсу информатики в соответствии с Федеральным государственным образовательным стандартом среднего (полного) общего образования (далее — ФГОС), который включает в себя источники:

- учебное пособие К.Ю. Полякова «Программирование. Python, C++», М.:, БИНОМ. Лаборатория знаний, 2018 г.;
- компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещённый на сайте: <http://kpolyakov.spb.ru/school/probook.htm>;
- электронный задачник-практикум с возможностью автоматической проверки решений задач по программированию: <http://informatics.mcsme.ru/course/view.php?id=666>;
- материалы для подготовки к итоговой аттестации по информатике в форме ОГЭ, размещённые на сайте материалы, размещенные на сайте <http://kpolyakov.spb.ru/school/oge.htm>;
- методическое пособие для учителя;
- комплект Федеральных цифровых информационно-образовательных ресурсов (далее ФЦИОР), помещенный в коллекцию ФЦИОР (<http://www.fcior.edu.ru>);
- сетевая методическая служба авторского коллектива для педагогов на сайте издательства <http://methodist.lbz.ru/authors/informatika/7/>;
- дистанционная подготовка:
 - Курс «Программирование на языке Python» на python «Д.П. Кириенко. «Программирование на языке Python» на языке Python (школа 179 г. Москва)» (<http://informatics.msk.ru/course/view.php?id=156>)
 - онлайн учебник Pythontutor.ru;
 - interactivepython.org — «Алгоритмы и структуры данных с использованием Python» (бесплатная книга с интерактивным тренажёром);
 - younglinux.info/oopython.php — Python. Введение в объектно-ориентированное программирование;
 - pygtk.org — библиотека PyGTK для разработки программ с графическим интерфейсом;
 - younglinux.info/tkinter.php — Tkinter. Программирование графического интерфейса;
 - effbot.org/tkinterbook/tkinter-index.htm — учебник по пакету tkinter;

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» включает в себя изучение ряда направлений в области программирования:

знакомство со средой программирования, этапами создания программ, программных модулей, овладение инструментами для тестирования и отладки программ.

Программа «Программирование на языке Python» имеет **техническую направленность**. Рассчитана на 1 год обучения и дает объем технических и естественно-научных компетенций, которыми вполне может овладеть современный школьник, ориентированный на научно-техническое и/или технологическое направление дальнейшего образования и сферу профессиональной деятельности. Возраст детей, участвующих в реализации программы 13-15 лет. Программа ориентирована, в первую очередь на ребят, желающих получить практические навыки программирования на высокоуровневом языке. Несмотря на то, что порог входа в курс достаточно низкий учащиеся должны обладать базовыми знаниями по алгоритмике и навыками составления программ для алгоритмических исполнителей на алгоритмическом языке. Знание других языков программирования не требуется.

Актуальность программы обусловлена потребностью общества в технически грамотных специалистах в области разработки прикладных программ, проектированию баз данных, машинного обучения и искусственного интеллекта. Знания, умения и практические навыки решения актуальных задач, полученные на занятиях по программированию, готовят учащихся к самостоятельной инженерной деятельности с применением современных технологий. В настоящее время существует дефицит IT-специалистов, в особенности в момент активного импортозамещения. Очень важную роль программирование, как метапредмет, играет в области образования, так как умение не просто составлять алгоритмы, но и оптимизировать их – важная компетенция современного конкурентоспособного инженера. Языки программирования встроены во многие профессиональные пользовательские пакеты используемые в разных сферах человеческой деятельности: Microsoft Office, Blender, 1С:Бухгалтерия. При помощи этих языков возможно максимально автоматизировать выполнение задач, тонко настроить процессы. Данные технологии поднимают образование на совершенно новый качественный уровень.

Глобальный запрос современного общества на виртуализацию нашей повседневной реальности требует подготовки большого количества инженеров-программистов по разным направлениям и предполагает повышенную заработную плату IT-специалистам, поэтому данное направление очень популярно и востребовано. Это закономерный этап в развитии общества. Очевидно, что он требует своевременного образования, обеспечивающего базу для естественного и осмысленного использования соответствующих устройств и технологий, профессиональной ориентации и обеспечения непрерывного образовательного процесса. Фактически программа призвана решить две взаимосвязанные задачи: профессиональная ориентация ребят в технически сложной сфере и формирование адекватного способа мышления.

Педагогическая целесообразность заключается не только в развитии технических способностей и возможностей средствами конструктивно-технологического подхода, гармонизации отношений ребенка и окружающего мира, но и в развитии созидательных способностей, устойчивого противостояния любым негативным социальным и социотехническим проявлениям. Движение в сторону нового технологического уклада требует воспитания в гражданах XXI века иных морально-этических качеств, наступает время, в которое решение множества естественных бытовых проблем ляжет на плечи искусственного разума, а значительная часть человеческой деятельности будет виртуализована. Данный процесс идет уже более 20 лет и связан с развитием и встраиванием в нашу жизнь сети Интернет. Так или иначе, но многие науки в своей методологии используют программирование, как способ автоматизации научной и исследовательской деятельности. Математика, физика, химия, лингвистика и прочие не могут развиваться дальше на скоростях обработки данных, которые были в прошлом. Для создания новых химических формул, прогнозирования процессов и явлений уже мало быстро вычисляющей техники, нужны современные оптимизированные алгоритмы с элементами машинного интеллекта. И именно в школе необходимо закладывать базу для дальнейшего формирования инженерных компетенций современного специалиста. Реальность такова, что те задачи, которые 40-50 лет назад решали

слушатели начальных курсов организаций высшего образования сегодня должен и может решать ученик 5-7 классов общеобразовательной школы.

Отличительной особенностью данной программы является то, что в её основе лежит идея использования в обучении собственной активности учащихся. Концепция данной программы - теория развивающего обучения в канве критического мышления. В основе сознательного акта учения в системе развивающего обучения лежит способность к продуктивному творческому воображению и мышлению. Более того, без высокого уровня развития этих процессов вообще невозможно ни успешное обучение, ни самообучение. Именно они определяют развитие творческого потенциала человека. Готовность к творчеству формируется на основе таких качеств как внимание и наблюдательность, воображение и фантазия, смелость и находчивость, умение ориентироваться в окружающем мире, произвольная память и др. Использование программы позволяет стимулировать способность детей к образному и свободному восприятию окружающего мира (людей, природы, культурных ценностей), его анализу и конструктивному синтезу.

Адресат программы.

Возраст детей, участвующих в реализации программы 13-15 лет.

Дети 13-15 лет, участвующие в реализации программы, это уже подростки. На смену конкретному приходит логическое мышление. Это проявляется в критицизме и требовании доказательств. Подросток теперь тяготеет к конкретным, его начинают интересовать философские вопросы (проблемы происхождения мира, человека). Происходит открытие мира психического, внимание подростка впервые обращается на других лиц. Для подростков характерно новое отношение к учению. Подросток стремится к самообразованию, причем часто становится равнодушным к оценке. Порой наблюдается расхождение между интеллектуальными возможностями и успехами в учебе: возможности высокие, а успехи низкие. Работая со учащимися, проявившими интерес к программированию незадолго до окончания 9-го класса, приходится особенно бережно и тщательно относиться к их времени: создавать индивидуальные задания, больше внимания уделять самостоятельной работе. При работе используются различные приемы групповой деятельности в разноуровневых группах для обучения элементам кооперации, внесения в собственную деятельность самооценки, взаимооценки, умение работать с технической литературой и выделять главное.

Объем программы и режим занятий.

Программа рассчитана на 1 год обучения. Общая продолжительность обучения составляет 72 часа.

Для успешной реализации программы целесообразно объединение детей в учебные группы численностью от 8 до 15 человек. В учебную группу принимаются все желающие. Но для успешного овладения программой ребята должны обладать базовыми знаниями по алгоритмике и навыками составления программ для алгоритмических исполнителей. Знание основ любого алгоритмического языка, позволит быстрее погрузиться в тему.

При определении режима занятий учтены санитарно-эпидемиологические требования к учреждениям дополнительного образования детей. Занятия проводятся 2 раза в неделю по 2 часа, с перерывом 15 мин (для детей старше 10 лет равна 45 мин (в соответствии с «Постановление Главного государственного санитарного врача РФ от 4 июля 2014 года №41 г. Москва «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству содержанию и организации режима работы образовательных организаций дополнительного образования детей»). Структура каждого занятия зависит от конкретной темы и решаемых в ней задач.

Формы и методы организации образовательного процесса.

Методика предусматривает проведение занятий в различных формах: групповой, парной, индивидуальной.

Групповые и парные занятия позволяют помочь ребёнку уверенно чувствовать себя в различных видах деятельности. Предполагается, что в течение полугода обучения у детей формируется достаточный уровень умений и навыков шаблонного проектирования. На этом фоне уже выделяются более компетентные, высоко мотивированные и даже, можно сказать, профессионально ориентированные дети.

Во втором полугодии рекомендуется индивидуальное проектирование, цель которого - развитие уникального сочетания способностей, умений и навыков и даже начальных профессиональных компетенций.

В рамках учебного плана каждого года особо выделены часы, используемые для разработки и подготовки к соревнованиям, участие в соревнованиях. Эти часы четко не распределены по времени, поскольку зависят от графика соревновательного процесса и результативности участия команд воспитанников.

Цель программы: формирование творческих и научно-технических компетенций обучающихся в неразрывном единстве с воспитанием коммуникативных качеств и целенаправленности личности через систему практико-ориентированных групповых занятий и самостоятельной деятельности обучающихся по программированию, решающих поставленные задачи.

Задачи программы.

Личностные:

- активизировать формирование активной жизненной позиции, гражданско-патриотической ответственности;
- воспитывать этику групповой работы, отношения делового сотрудничества, взаимоуважения;
- развивать основы коммуникативных отношений внутри проектных групп и в коллективе в целом;
- воспитывать упорство в достижении результата;
- пропагандировать здоровый образ жизни;
- формировать целеустремлённость, организованность, равнодушие, ответственное отношение к труду и уважительное отношение к окружающим.

Метапредметные:

- сформировать и развить креативность, гибкость и самостоятельность мышления на основе игровых образовательных и воспитательных технологий;
- сформировать и развить навыки проектирования и построения алгоритмов;
- создать оптимальное мотивационное пространство для детского творчества.

Предметные:

- дать представление об основных понятиях программирования, разных парадигмах программирования;
- дать представление о классических алгоритмах и их свойствах;
- дать представление о специфике интерпретируемых программ, их преимуществах и недостатках, отличиях от компилируемых программ;
- дать представление о технических характеристиках оборудования для аккуратного использования памяти компьютера, возможностей процессора;
- познакомить с авторским правом, с лицензированием программного обеспечения;
- развить навыки программирования в современной среде программирования углубить знания, повысить мотивацию к обучению путем практического интегрированного применения знаний, полученных в различных образовательных областях (математика, физика, информатика);
- развить интерес к научно-техническому, инженерно-конструкторскому творчеству, сформировать общенаучные и технологические навыки проектирования, развить творческие способности учащихся.

Содержание программы

Первый уровень (17 часов).

- Простейшие программы. Диалоговые программы. Переменные. Консольный ввод и вывод данных.
- Компьютерная графика. Система координат. Управление пикселями. Графические примитивы: линии, прямоугольники, окружности. Изменение координат. Анимация.
- Процедуры. Процедуры с параметрами. Рефакторинг.
- Обработка целых чисел. Арифметические выражения. Деление нацело. Остаток от деления.
- Обработка вещественных чисел. Особенности представления вещественных чисел в памяти компьютера. Операции с вещественными числами.
- Случайные и псевдослучайные числа. Генераторы случайных чисел.
- Ветвления. Условный оператор. Полная и неполная формы условного оператора. Вложенные условные операторы. Логические переменные. Экспертные системы. Сложные условия. Логические операции И, ИЛИ, НЕ. Порядок выполнения операций.
- Циклы с условием. Алгоритм Евклида. Обработка потока данных. Бесконечные циклы. Циклы по переменной. Шаг изменения переменной цикла.
- Циклы в компьютерной графике. Узоры. Вложенные циклы. Штриховка.

Второй уровень (18 часов).

- Этапы создания программ. Методы проектирования программ «сверху вниз» и «снизу вверх». Интерфейс и реализация. Документирование программы.
- Подпрограммы: процедуры и функции. Процедуры. Процедуры с параметрами. Локальные и глобальные переменные.
- Функции. Логические функции.
- Рекурсия. Рекурсивные процедуры и функции. Фракталы.
- Символьные строки. Сравнение строк. Операции со строками. Обращение к символам. Перебор всех символов. Срезы. Удаление и вставка. Встроенные методы. Поиск в символьных строках. Замена символов. Преобразования «строка — число». Символьные строки в функциях. Рекурсивный перебор.
- Массивы (списки). Массивы в языке Python. Создание массива. Обращение к элементу массива. Перебор элементов массива. Генераторы. Вывод массива. Ввод массива с клавиатуры. Заполнение массива случайными числами.
- Алгоритмы обработки массивов. Сумма элементов массива. Подсчёт элементов массива, удовлетворяющих условию. Особенности копирования списков в Python.
- Поиск в массивах. Линеиный поиск. Поиск максимального элемента в массиве. Максимальный элемент, удовлетворяющий условию. Использование массивов в прикладных задачах.
- Матриц. Создание и заполнение матриц. Вывод матрицы на экран. Перебор элементов матрицы. Квадратные матрицы.
- Сложность алгоритмов. Асимптотическая сложность.

Третий уровень (18 часов).

- Сортировка массивов. Метод пузырька (сортировка обмeнами). Метод выбора. Сортировка слиянием. Быстрая сортировка. Сортировка в языке Python.
- Двоичный поиск в массиве данных. Двоичный поиск по ответу.
- Обработка файлов. Типы файлов. Чтение данных. Запись данных. Обработка данных из файла.
- Целочисленные алгоритмы. Решето Эратосфена. Целочисленный квадратный корень.
- Словари. Алфавитно-частотный словарь. Перебор элементов словаря.
- Структуры. Классы. Создание структур. Работа с полями структур. Хранение структур в файлах. Сортировка структур.
- Стек. Использование списка. Вычисление арифметических выражений. Скобочные выражения. Системный стек. Очередь. Дек.
- Деревья. Деревья поиска. Обход дерева. Использование связанных структур. Вычисление арифметических выражений.
- Графы. Описание графа. Жадные алгоритмы. Минимальное остовное дерево. Алгоритм Дейкстры. Алгоритм Флойда–Уоршелла. Использование списков смежности.
- Динамическое программирование. Числа Фибоначчи. Количество программ для исполнителя. Двумерные задачи. Поиск оптимального решения.
- Игровые модели. Выигрышные и проигрышные позиции.

Четвёртый уровень (18 часов).

- Проблема сложности программ. Процедурный и объектно-ориентированный подходы к написанию программ.
- Классы и объекты. Объектно-ориентированный анализ. Взаимодействие объектов. Свойства и методы.
- Классы и объекты в программе. Объявление класса. Поля класса. Конструктор класса. Данные и методы класса.
- Скрытие внутреннего устройства. Доступ к полям через методы. Свойства (property). Свойство «только для чтения»
- Иерархия классов. Наследование. Базовый класс. Доступ к полям. Классы-наследники. Полиморфизм. Разработка модулей.
- Событийно-ориентированное программирование. Программы с графическим интерфейсом. Форма. Свойства формы. Обработчики событий.
- Использование компонентов (виджетов). Ввод и вывод данных. Обработка ошибок с помощью исключений.
- Создание компонентов. Добавление свойств и методов. Составные компоненты.
- Модель и представление.

1 уровень

№	Тема занятия	Параграф пособия (номер, название)	Кол-во часов	
			теория	практ.
1.	Первые программы	§ 1. Первые программы	0,5	0,5
2.	Диалоговые программы	§ 2. Диалоговые программы	0,5	0,5
3.	Компьютерная графика	§ 3. Компьютерная графика	0,5	0,5
4.	Процедуры	§ 4. Процедуры	0,5	0,5
5.	Обработка целых чисел	§ 5. Обработка целых чисел	0,5	0,5
6.	Обработка вещественных чисел	§ 6. Обработка вещественных чисел	0,5	0,5
7.	Случайные и псевдослучайные числа	§ 7. Случайные и псевдослучайные числа	0,5	0,5
8.	Ветвления	§ 8. Ветвления	0,5	0,5
9.	Сложные условия	§ 9. Сложные условия	0,5	0,5
10.	Циклы с условием	§ 10. Циклы с условием	0,5	0,5
11.	Циклы с условием: практикум	§ 10. Циклы с условием		1
12.	Анимация	§ 11. Анимация	0,5	0,5
13.	Циклы по переменной	§ 12. Циклы по переменной	0,5	0,5
14.	Циклы в компьютерной графике	§ 13. Циклы в компьютерной графике	0,5	0,5
15.	Выполнение проекта	§ 1-13.		1
16.	Выполнение проекта	§ 1-13.		1
17.	Выполнение проекта	§ 1-13.		1
Итого			6,5	10,5

2 уровень

№	Тема занятия	Параграф пособия (номер, название)	Кол-во часов	
			теория	практ.
1.	Проектирование программ	§ 1. Проектирование программ	1	
2.	Процедуры	§ 2. Процедуры	0,5	0,5
3.	Рекурсия	§ 3. Рекурсия	0,5	0,5
4.	Функции	§ 4. Функции	0,5	0,5
5.	Символьные строки	§ 5. Символьные строки	0,5	0,5
6.	Обработка символьных строк	§ 5. Символьные строки	0,5	0,5
7.	Строки в функциях	§ 5. Символьные строки	0,5	0,5
8.	Массивы	§ 6. Массивы (списки)	0,5	0,5
9.	Ввод и вывод массивов	§ 6. Массивы (списки)	0,5	0,5
10.	Суммирование элементов массива	§ 7. Алгоритмы обработки массивов	0,5	0,5
11.	Подсчёт элементов массива, удовлетворяющих условию	§ 7. Алгоритмы обработки массивов	0,5	0,5
12.	Поиск значения в массиве	§ 8. Поиск в массивах	0,5	0,5
13.	Поиск максимального элемента в массиве	§ 8. Поиск в массивах	0,5	0,5
14.	Игра «Стрельба по тарелкам»	§ 9. Используем массивы	0,5	0,5
15.	Игра «Стрельба по тарелкам»	§ 9. Используем массивы	0,5	0,5
16.	Матрицы	§ 10. Матрицы	0,5	0,5
17.	Сложность алгоритмов	§ 11. Сложность алгоритмов	1	
18.	Итоговый проект			1
Итого			9,5	8,5

3 уровень

№	Тема занятия	Параграф пособия (номер, название)	Кол-во часов	
			теория	практ.
1.	Простые алгоритмы сортировки	§ 1. Простые алгоритмы сортировки	0,5	0,5
2.	Сортировка слиянием	§ 2. Быстрые алгоритмы сортировки	0,5	0,5
3.	Быстрая сортировка	§ 2. Быстрые алгоритмы сортировки	0,5	0,5
4.	Двоичный поиск	§ 3. Двоичный поиск	0,5	0,5
5.	Обработка файлов	§ 4. Обработка файлов	0,5	0,5
6.	Обработка файлов: практикум	§ 4. Обработка файлов		1
7.	Целочисленные алгоритмы	§ 5. Целочисленные алгоритмы	0,5	0,5

№	Тема занятия	Параграф пособия (номер, название)	Кол-во часов	
			теория	практ.
8.	Словари	§ 6. Словари	0,5	0,5
9.	Структуры	§ 7. Структуры	0,5	0,5
10.	Структуры: практикум	§ 7. Структуры		1
11.	Стек, очередь, дек	§ 8. Стек, очередь, дек	0,5	0,5
12.	Деревья	§ 9. Деревья	0,5	0,5
13.	Графы	§ 10. Графы	0,5	0,5
14.	Графы: практикум	§ 10. Графы		1
15.	Динамическое программирование	§ 11. Динамическое программирование	0,5	0,5
16.	Динамическое программирование: практикум	§ 11. Динамическое программирование	0,5	0,5
17.	Игровые модели	§ 12. Игровые модели	0,5	0,5
		Итого	7	10

4 уровень

№	Тема занятия	Параграф пособия (номер, название)	Кол-во часов	
			теория	практ.
1.	Что такое ООП?	§ 1. Что такое ООП?	1	
2.	Модель задачи: классы и объекты	§ 2. Модель задачи: классы и объекты	0,5	0,5
3.	Классы и объекты в программе	§ 3. Классы и объекты в программе	0,5	0,5
4.	Классы и объекты в программе: практикум	§ 3. Классы и объекты в программе		1
5.	Скрытие внутреннего устройства	§ 4. Скрытие внутреннего устройства	0,5	0,5
6.	Иерархия классов	§ 5. Иерархия классов	0,5	0,5
7.	Классы-наследники (I)	§ 6. Классы-наследники (I)	0,5	0,5
8.	Классы-наследники (II)	§ 7. Классы-наследники (II)	0,5	0,5
9.	Доработка игры	§ 5-7.		1
10.	Событийно-ориентированное программирование	§ 8. Событийно-ориентированное программирование	0,5	0,5
11.	Использование компонентов (виджетов)	§ 9. Использование компонентов (виджетов)	0,5	0,5
12.	Использование компонентов (виджетов)	§ 9. Использование компонентов (виджетов)	0,5	0,5
13.	Создание компонентов	§ 10. Создание компонентов	0,5	0,5
14.	Модель и представление	§ 11. Модель и представление	0,5	0,5
15.	Выполнение проекта	§ 1-11.		1
16.	Выполнение проекта	§ 1-11.		1
17.	Выполнение проекта	§ 1-20.		1
18.	Выполнение проекта	§ 1-20.		1
		Итого	6,5	11,5

Планируемые результаты

Программа разделена на четыре уровня, ассоциируемых со «ступенями» повышения компетенций учащегося.

Первый уровень

В результате изучения курса на первом уровне учащийся

- 1) научится составлять и отлаживать простые диалоговые программы;
- 2) узнает особенностей машинных вычислений с целыми и вещественными числами;
- 3) научится использовать основные алгоритмические конструкции: условные операторы, циклы с условием, циклы по переменной;

- 4) овладеет методами построения графических изображений программными средствами;
- 5) овладеет простыми методами программирования компьютерной анимации.

Второй уровень

В результате изучения курса на втором уровне учащийся

- 6) познакомится с методами проектирования программ «сверху вниз» и «снизу вверх»;
- 7) научится использовать вспомогательные алгоритмы (процедуры и функции) для структуризации программ;
- 8) научится применять рефакторинг для улучшения читаемости программ;
- 9) научится использовать символьные строки;
- 10) овладеет основными алгоритмами обработки одномерных и двумерных массивов;
- 11) познакомится с понятием сложности алгоритма;

Третий уровень

В результате изучения курса на третьем уровне учащийся

- 12) научится применять различные алгоритмы сортировки массивов;
- 13) научится использовать двоичный поиск;
- 14) научится обрабатывать данные, записанные в текстовые и двоичные файлы, и сохранять в файлах результаты работы программы;
- 15) научится использовать структуры для объединения данных;
- 16) научится применять словари, стеки, очереди, деки для решения задач обработки данных;
- 17) научится использовать деревья для организации данных;
- 18) познакомится с методами описания графов и некоторыми популярными алгоритмами на графах;
- 19) научится использовать динамическое программирование для решения комбинаторных и оптимизационных задач;
- 20) познакомится с понятием выигрышных и проигрышных позиций в играх с полной информацией;

Четвёртый уровень

В результате изучения курса на четвёртом уровне учащийся

- 21) познакомится с объектно-ориентированным подходом к разработке программ;
- 22) научится выполнять объектно-ориентированный анализ задачи, выделять свойства и методы объектов;
- 23) научится использовать инкапсуляцию для защиты данных объектов;
- 24) познакомится с понятиями «класс» и «абстрактный класс»;
- 25) познакомится с понятиями «инкапсуляция», «наследование», «полиморфизм»;
- 26) научится проектировать несложные иерархии классов для прикладных задач;
- 27) познакомится с принципами разработки событийно-ориентированных программ;
- 28) научится создавать программы с графическим интерфейсом на языках Python и C#;

29) научиться использовать готовые и создавать новые компоненты (виджеты) для сред быстрой разработки программ.

Комплекс организационно-педагогических условий.

Для реализации программы «Программирование на языке Python» помещение должно соответствовать следующим характеристикам.

Учебный класс соответствует требованиям санитарных норм и правил, установленных СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству содержанию и организации режима работы образовательных организаций дополнительного образования детей», утвержденных Постановлением Главного государственного санитарного врача Российской Федерации от 4 июля 2014 г. №41.

Класс оборудован учебной мебелью: рабочее место учителя, рабочие места учащихся для работы с теоретическим материалом и с оборудованием для практических работ, компьютерные столы для стационарно установленных компьютеров (компьютерный класс), классная доска, шкаф для хранения электронных компонентов и дополнительного оборудования.

Технические средства обучения включают компьютер преподавателя, акустическую систему, мультимедийный проектор, мобильный компьютер (ноутбук), индивидуальный, на каждого обучающегося.

Специализированное программное обеспечение, которое может быть использовано для реализации данной программы, имеет либо открытые лицензии, либо бесплатные, либо предоставляется образовательным организациям в рамках различных образовательных программ. Состав:

- Thonny;
- Visual Studio Code;
- Eclipse;
- Python 3+.

Специальных требований к одежде обучающихся нет, за исключением регламентированных правилами образовательного учреждения.

Формы аттестации

Текущий контроль - проводится по окончании изучения темы в виде устного опроса, практической работы, через просмотры работ, при этом оцениваются усвоение и качество выполнения изучаемых на занятиях приемов и операций, выявление ошибок и успехов в работе.

Промежуточная аттестация – проводится за каждое полугодие по пройденным темам, осуществляется при помощи практических заданий и устного опроса по теории. При оценке результатов также учитывается качество выполненных работ, уровень творческой деятельности, найденные продуктивные технические и технологические решения, степень самостоятельности. Мониторинг развития качеств личности учащихся проводится в конце учебного года по таким качествам личности как активность, организаторские способности; коммуникативные навыки, коллективизм; ответственность, самостоятельность, дисциплинированность; нравственность, гуманность; креативность, склонность к исследовательско-проектировочной деятельности.

Оценивание результатов обучения

Результатом обучения считается способность учащегося написать программу (разработать проект) определённого уровня сложности. Далее выделяются следующие уровни сложности:

А: начальный уровень, воспроизведение изучаемого материала с незначительными изменениями;

В: средний уровень, способно применять изученный материал для написания программ, которые отличаются от изученных;

С: высокий уровень, способно применять изученный материал для самостоятельного написания программ, решающих нестандартные задачи.

Глава 1. Программирование на языке Python

§ 1. Первые программы

§ 2. Диалоговые программы

А: Напишите программу, которая вводит три целых числа и находит их сумму.

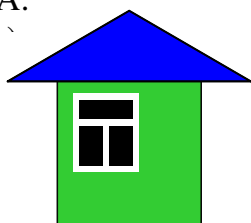
В: Напишите программу, которая вводит три целых числа и находит их сумму и произведение.

С: Напишите программу, которая вводит три целых числа и находит их сумму, произведение и среднее арифметическое.

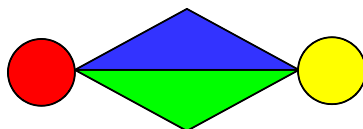
§ 3. Компьютерная графика

А-С: Напишите программу, которая строит следующий рисунок:

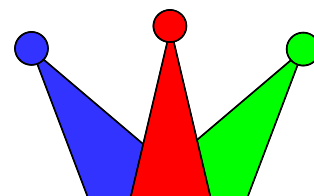
А:



В:



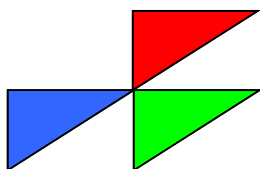
С:



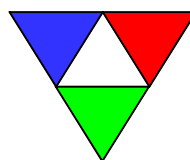
§ 4. Процедуры

А-С: Напишите программу, которая строит следующий рисунок, используя единственную процедуру:

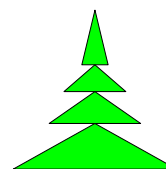
А:



В:



С:



§ 5. Обработка целых чисел

А: Напишите программу, которая вводит трёхзначное число и разбивает его на цифры.

В: Напишите программу, которая вводит с клавиатуры количество секунд и выводит то же самое время в часах, минутах и секундах.

С: Напишите программу, которая вводит с клавиатуры четырёхзначное натуральное число и переставляет его первую и последнюю цифры, например, из числа 1234 должно получиться число 4231.

§ 6. Обработка вещественных чисел

А: Напишите программу, которая вычисляет квадратный корень введённого числа.

В: Напишите программу, которая вводит с клавиатуры координаты двух точек на плоскости и вычисляет длину соединяющего их отрезка.

С: Напишите программу, которая вводит с клавиатуры координаты двух точек в пространстве и вычисляет длину соединяющего их отрезка.

§ 7. Случайные и псевдослучайные числа

- А:** В игре «Русское лото» из мешка случайным образом выбираются бочонки, на каждом из которых написано число от 1 до 90. Напишите программу, которая выводит наугад первые 5 выигрышных номеров.
- В:** Напишите программу, которая моделирует бросание двух игральных кубиков: при запуске выводит случайное число на отрезке $[2; 12]$.
- С:** Игральный кубик бросается три раза (выпадает три случайных значения). Из этих чисел составляется целое число, программа должна найти его квадрат.

§ 8. Ветвления

- А:** Напишите программу, которая вводит три целых числа и находит максимальное из них.
- В:** Напишите программу, которая вводит пять целых чисел и находит максимальное из них.
- С:** Напишите программу, которая вводит последовательно возраст Антона, Бориса и Виктора и определяет, кто из них старше.

§ 9. Сложные условия

- А:** Напишите программу, которая получает три числа и выводит количество одинаковых чисел в этой цепочке.
- В:** Напишите программу, которая получает номер месяца и выводит соответствующее ему время года или сообщение об ошибке.
- С:** Напишите программу, которая получает возраст человека (целое число, не превышающее 120) и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «25 лет».

§ 10. Циклы с условием

- А:** Напишите программу, которая получает два целых числа A и B ($0 < A < B$) и выводит квадраты всех натуральных чисел на отрезке $[A, B]$.
- В:** Напишите программу, которая получает два целых числа и находит их произведение, не используя операцию умножения. Числа могут быть отрицательными.
- С:** Напишите программу, которая вводит натуральное число N и вычисляет сумму всех чисел Фибоначчи, меньших N . Предусмотреть защиту от ввода отрицательного числа N .

§ 11. Анимация

- А:** Напишите программу, которая моделирует движение объекта, пролетающего через игровое поле.
- В:** Напишите программу, которая моделирует движение двух объекта, пролетающих через игровое поле в разных направлениях.
- С:** Напишите программу, которая моделирует движение двух объекта, пролетающих через игровое поле в разных направлениях. Объекты должны отталкиваться от стенок поля.

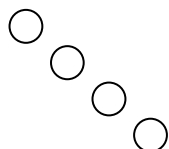
§ 12. Циклы по переменной

- А:** Напишите программу, которая находит все пятизначные числа, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.
- В:** Напишите программу, которая находит все трёхзначные Армстронга (для которых сумма кубов цифр числа равна самому числу).
- С:** Напишите программу, которая получает натуральное число N и выводит на экран все автоморфные числа, не превосходящие N (натуральное число называется автоморфным, если его десятичная запись совпадает с последними цифрами его квадрата, например, $25^2 = 625$).

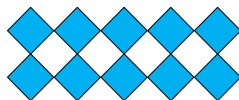
§ 13. Циклы в компьютерной графике

- А-С:** Напишите программу, которая строит узор:

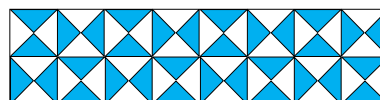
а)



б)



в)



Второй уровень

Глава 1. Программирование на языке Python

§ 1. Проектирование программ

§ 2. Процедуры

А: Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран квадрат из звёздочек размером $N \times N$ символов.

В: Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с последней.

С: Напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления.

§ 3. Рекурсия

А: Напишите рекурсивную процедуру, которая переводит число в троичную систему счисления.

В: Напишите рекурсивную процедуру, которая переводит число в шестнадцатеричную систему счисления.

С: Напишите рекурсивную процедуру, которая переводит число в негавдоичную систему счисления (систему с основанием -2).

§ 4. Функции

А: Напишите функцию, которая возвращает старшую цифру переданного ей натурального числа.

В: Напишите функцию, которая возвращает количество делителей переданного ей натурального числа.

С: Напишите логическую функцию, которая возвращает результат *True*, если переданное ей число – это число Фибоначчи.

§ 5. Символьные строки

А: Напишите программу, которая заменяет в символьной строке все точки на нули и все буквы «X» на единицы.

В: Напишите программу, которая выполняет инверсию битовой строки: заменяет в ней все нули на единицы и наоборот.

С: Напишите программу, которая вычисляет сумму неизвестного количества натуральных чисел, записанную в виде символьной строки, например, «1+25+12+34+89»

§ 6. Массивы (списки)

А: Напишите программу, которая вводит целое число X и заполняет массив последовательными натуральными числами, начиная с X .

В: Напишите программу, которая вводит целое число X и заполняет массив последовательными натуральными числами, начиная с X , в обратном порядке.

С: Напишите программу, которая заполняет массив из N элементов первыми N числами Фибоначчи.

§ 7. Алгоритмы обработки массивов

- А:** Напишите программу, которая определяет сумму элементов массива с чётными значениями.
- В:** Напишите программу, которая в массиве с чётным количеством элементов меняет местами пары соседних элементов.
- С:** Напишите программу, которая находит в массиве все числа Фибоначчи и строит из них новый массив.

§ 8. Поиск в массивах

- А:** Напишите программу, которая находит в массиве минимальный и максимальный элементы и их индексы.
- В:** Напишите программу, которая находит в массиве количество элементов, равных минимальному, и их индексы.
- С:** Напишите программу, которая находит в массиве два максимальных элемента и их индексы.

§ 9. Используем массивы

- А:** Закончите программу из параграфа.
- В:** Доработайте программу из параграфа так, чтобы тарелки двигались в другом направлении.
- С:** Доработайте программу из параграфа так, чтобы попадание в меньшую по размеру тарелку приводило к большему увеличению счёта.

§ 10. Матрицы

- А:** Напишите программу, которая находит максимальный элемент на главной диагонали квадратной матрицы.
- В:** Напишите программу, которая находит максимальный элемент матрицы и его индексы (номера строки и столбца).
- С:** Напишите программу, которая выполняет транспонирование матрицы.

§ 11. Сложность алгоритмов

Контроль не предусмотрен.

Третий уровень

Глава 1. Программирование на языке Python

§ 1. Простые алгоритмы сортировки

- А:** Напишите программу, которая выполняет сортировку массива методом «пузырька».
- В:** Напишите программу, которая выполняет сортировку массива методом «пузырька» в обратном порядке (сверху вниз).
- С:** Напишите программу, которая выполняет сортировку двух половин массива (отдельно) методом выбора.

§ 2. Быстрые алгоритмы сортировки

- А:** Напишите программу, которая выполняет сортировку массива по убыванию методом слияния.
- В:** Напишите программу, которая выполняет быструю сортировку массива символьных строк по убыванию длины.
- С:** Напишите программу, которая сравнивает количество операций при сортировке массива методами «пузырька» и слияния.

§ 3. Двоичный поиск

- А:** Напишите программу, которая находит в отсортированном массиве индексы всех элементов, равных заданному значению X .

В: Напишите программу, которая сравнивает количество проверок при линейном и двоичном поиске в отсортированном массиве.

С: Напишите программу, которая определяет среднее количество проверок при двоичном поиске в отсортированном массиве. Используя результаты работы этой программы, постройте зависимости количества проверок от размера массива.

§ 4. Обработка файлов

А: Напишите программу, которая вычисляет сумму чисел, записанных в файле в столбик.

В: Напишите программу, которая сортирует числа, записанные в файле в столбик.

С: Напишите программу, которая сортирует набор чисел, записанных в двух файлах в столбик. Отсортированные числа должны быть записаны в новый файл.

§ 5. Целочисленные алгоритмы

А: Напишите программу, которая вводит натуральное число N и находит все простые числа на отрезке $[2; N]$. Используйте алгоритм «решето Эратосфена».

В: Напишите программу, которая вводит натуральные числа M и N и находит все простые числа на отрезке $[M; N]$. Используйте алгоритм «решето Эратосфена».

С: Напишите программу, которая определяет количество операций, которые выполняются при решении задачи А. Используя результаты работы этой программы, постройте зависимость количества операций (или времени работы программы) от N .

§ 6. Словари

А: Напишите программу, которая строит алфавитно-частотный словарь для заданного файла, в котором каждое слово записано в отдельной строке.

В: Напишите программу, которая строит алфавитно-частотный словарь для заданного файла с произвольным текстом.

С: Напишите программу, которая сравнивает тексты разных авторов на основе частоты использования предлогов и союзов.

§ 7. Структуры

А: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление и просмотр записей.

В: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление, просмотр и поиск записей по ключу.

С: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление, просмотр, поиск записей по ключу и сортировку по ключу.

§ 8. Стек, очередь, дек

А: Напишите программу, которая проверяет правильность скобочного выражения с тремя типами скобок.

В: Напишите программу, которая вычисляет значение арифметического выражения, записанного в префиксной форме.

С: Напишите программу, которая применяет алгоритм заливки области для поиска пути в лабиринте.

§ 9. Деревья

А: Напишите программу, которая вычисляет введённое арифметическое выражение без скобок.

В: Напишите программу, которая вычисляет введённое арифметическое выражение со скобками.

С: Напишите программу, которая вычисляет введённое арифметическое выражение со скобками и функциями (\sin , \cos , $\sqrt{\quad}$, abs).

§ 10. Графы

- А:** Напишите программу, которая строит минимальное остовное дерево для неориентированного графа.
- В:** Напишите программу, которая определяет кратчайший маршрут из одной вершины графа в другую с помощью алгоритма Дейкстры.
- С:** Напишите программу, которая определяет кратчайшие маршрут между каждой парой вершин графа с помощью алгоритма Флойда-Уоршелла.

§ 11. Динамическое программирование

- А:** Напишите программу, которая вычисляет количество битовых цепочек длины N , в которых нет двух единиц подряд.
- В:** Напишите программу, которая вычисляет количество битовых цепочек длины N , в которых нет трёх единиц подряд.
- С:** Напишите программу, которая решает задачу о ранце.

§ 12. Игровые модели

- А:** Напишите программу, которая играет с человеком в игру Баше.
- В:** Напишите программу, которая определяет тип позиции (выигрышная или проигрышная) в игре «крестики-нолики» на доске 3×3 .
- Напишите программу, которая играет с человеком в «крестики-нолики».

Четвёртый уровень

Глава 1. Программирование на языке Python

§ 1. Что такое ООП?

§ 2. Модель задачи: классы и объекты

- А:** Перечислите свойства и методы, которыми могли бы обладать объекты класса *Яхта*.
- В:** Постройте объектно-ориентированную модель для задачи моделирования дорожного движения. Рассмотрите классы объектов *Дорога*, *Светофор*, *Машина* и их взаимосвязь.
- С:** Постройте объектно-ориентированную модель для придуманной вами задачи моделирования.

§ 3. Классы и объекты в программе

- А:** Закончите программу, рассмотренную в параграфе.
- В:** Добавьте в программу из параграфа класс *Торпедный аппарат*.
- С:** Добавьте в программу из параграфа классы *Торпедный аппарат* и *Торпеда*.

§ 4. Скрытие внутреннего устройства

- А:** Постройте программную реализацию класса *Перо*. Все данные должны быть скрыты.
- В:** Постройте программную реализацию класса *Машина*. Все данные должны быть скрыты.
- С:** Измените программу игры «Торпедная атака» так, чтобы все поля у объектов были закрытыми.

§ 5. Иерархия классов

- А:** В программе нужно моделировать следующие классы: *Озеро*, *Пресное озеро*, *Солёное озеро*, *Река*, *Море*, *Водоём*, *Океан*. Постройте иерархию классов для этой задачи.
- В:** В программе нужно моделировать следующие классы: *Дерево*, *Осина*, *Берёза*, *Озеро*, *Река*, *Заяц*, *Волк*, *Медведь*. Постройте иерархию классов для этой задачи.
- С:** В программе нужно моделировать следующие классы: *Корабль*, *Подводная лодка*, *Самолет*, *Вертолет*, *Гидросамолет*, *Мотоцикл*, *Трактор*. Постройте иерархию классов для этой задачи.

§ 6. Классы-наследники (I)

- А:** Закончите программу из параграфа.

В: Добавьте в программу из задания *A* ещё один класс неподвижных объектов, которые изображаются квадратом или ромбом.

С: Добавьте в программу из задания *A* ещё один класс неподвижных объектов, которые при анимации изменяются случайным образом.

§ 7. Классы-наследники (II)

А: Закончите программу из текста параграфа.

В: Измените программу так, чтобы космический корабль, который попал в чёрную дыру или в пульсар, уничтожился. Вместо него должен появляться новый корабль в случайном месте.

С: Добавьте в игру объекты класса *TDestroyer* – специальные боевые космические корабли, которые охотятся за кораблями и странниками и уничтожают их при встрече.

§ 8. Событийно-ориентированное программирование

А: Постройте программу, которая запрашивает разрешение на завершение работы.

В: Измените цвет формы в программе из задания *A*.

С: Доработайте программу из задания *A* так, чтобы при щелчке мышью по форме изменялся цвет формы.

§ 9. Использование компонентов (виджетов)

А: Закончите программу для работы с RGB-кодами цвета.

В: Напишите программу для перевода чисел из восьмеричной системы счисления в десятичную.

С: Напишите программу для перевода чисел из десятичной системы счисления в римскую и обратно.

§ 10. Создание компонентов

А: Постройте компонент для ввода целых чисел.

В: Постройте компонент для ввода целых чисел в шестнадцатеричной системе счисления.

С: Постройте компонент для ввода чисел в римской системе счисления.

§ 11. Модель и представление

А: Закончите программу из параграфа.

В: Постройте программу «Калькулятор» для выполнения вычислений с целыми числами.

С: Постройте программу «Калькулятор» для выполнения вычислений с целыми числами в римской системе счисления.

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 603332450510203670830559428146817986133868575796

Владелец Пилипушка Сергей Владимирович

Действителен с 25.03.2022 по 25.03.2023