

РАБОЧАЯ ПРОГРАММА
учебного предмета «Информатика: линейное программирование»
для обучающихся 11 классов

Содержание

Пояснительная записка	3
1.Содержание учебного предмета «Информатика: линейное программирование»	5
11 класс	5
2.Планируемые результаты освоения программы учебного предмета «Информатика: линейное программирование»	13
2.1 Личностные результаты	13
2.2 Метапредметные результаты	14
2.3. Предметные результаты	16
3.Тематическое планирование	
11 класс	20

Пояснительная записка

Элективный учебный предмет «Информатика: линейное программирование» составлена на базе программы Полякова Константина Юрьевича «Программирование на языках Python и C++».

В мире высоких технологий программирование является неотъемлемой частью. Компьютеры уже давно вошли в нашу жизнь, и любой человек, которому приходится хоть какое-то время проводить за компьютером, так или иначе, сталкивается с программированием. В свою очередь программирование базируется на языках программирования, которых на сегодняшний день насчитывают огромное множество. Python – это современный универсальный язык программирования, с помощью которого можно создавать любые приложения в диапазоне от интернет-сайтов до роботов и системных сервисов. Его достоинства: – кроссплатформенность и бесплатность; – простой синтаксис и богатые возможности позволяют записывать программы очень кратко, но в то же время понятно; – по простоте освоения язык сравним с бейсиком, но куда более богат возможностями и значительно более современен; – богатая стандартная библиотека, возможность разработки промышленных приложений (для работы с сетью, GUI, базами данных и т.д.) Главная идея элективного учебного предмета «Язык Python: основы программирования» – это организация систематического и системного повторения, углубления и расширения знаний по теме «алгоритмы и элементы программирования» курса информатики за период изучения в основной школе. Вопросы, рассматриваемые в элективном учебном предмете, выходят за рамки обязательного содержания. Вместе с тем, они тесно примыкают к основному курсу информатики, поэтому данный элективный учебный предмет будет способствовать совершенствованию и развитию важнейших знаний и умений в области информатики, предусмотренных школьной программой. Данный элективный учебный предмет является практико-ориентированным, дает учащимся возможность познакомиться с основами программирования на языке Python и применить знания на практике, также позволяет успешно готовиться к участию в олимпиадах, конкурсах.

Программа составлена с учетом возрастных особенностей, учащихся старшего школьного возраста и предусмотрена на работу в компьютерном классе, рассчитана на 1 час в неделю.

Основной целью элективного учебного предмета является формирование у учащихся навыков алгоритмического и логического стиля мышления, представления о приемах и методах программирования через составление алгоритмов и обучение искусству программирования.

В соответствии с поставленной целью можно выделить следующие задачи:

образовательные:

– способствовать формированию учебно-интеллектуальных умений, приёмов мыслительной деятельности, освоению рациональных способов её осуществления на основе учета индивидуальных особенностей учащихся;

– способствовать формированию активного, самостоятельного, креативного мышления; – научить основным приемам и методам программирования.

развивающие:

– развивать психические познавательные процессы: мышление, восприятие, память, воображение у учащихся;

– развивать представление учащихся о практическом значении информатики.

воспитательные:

– воспитывать культуру алгоритмического мышления;

– воспитывать у учащихся усидчивость, терпение, трудолюбие.

Данный элективный учебный предмет имеет прикладное и общеобразовательное значение, способствует развитию логического мышления учащихся, систематизации знаний при подготовке к выпускным экзаменам. Используются различные формы организации занятий, такие как лекция и семинар, групповая, индивидуальная деятельность учащихся.

Преобладающий тип занятий – практикум. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения практических работ.

Элективный учебный предмет «Информатика: линейное программирование» изучается в 11 классе как предмет по выбору в объёме 1 часа в неделю, всего 34 часа.

Содержание учебного предмета, курса

11 КЛАСС (34 часа)

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.

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. Линейный 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.

Сортировка массивов. Метод пузырька (сортировка обменами). 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. 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.

Планируемые результаты освоения учебного предмета, курса

В результате изучения курса учащийся

1. научиться составлять и отлаживать простые диалоговые программы;
2. узнает особенностей машинных вычислений с целыми и вещественными числами;
3. научиться 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.
6. познакомится с методами проектирования программ «сверху вниз» и «снизу вверх»;
7. научиться использовать вспомогательные алгоритмы (процедуры и функции) для структуризации программ;
8. научиться применять рефакторинг для улучшения читаемости программ;
9. научиться использовать символьные строки;
10. овладеет основными алгоритмами обработки одномерных и двумерных массивов;
11. познакомится с понятием сложности алгоритма;
12. научиться применять различные алгоритмы сортировки массивов;
13. научиться использовать двоичный поиск;
14. научиться обрабатывать данные, записанные в текстовые и двоичные файлы, и сохранять в файлах результаты работы программы;
15. научиться использовать структуры для объединения данных;
16. научиться применять словари, стеки, очереди, деки для решения задач обработки данных;
17. научиться использовать деревья для организации данных;
18. познакомится с методами описания графов и некоторыми популярными алгоритмами на графах;
19. научиться использовать динамическое программирование для решения комбинаторных и оптимизационных задач;
20. познакомится с понятием выигрышных и проигрышных позиций в играх с полной информацией;
21. познакомится с объектно-ориентированным подходом к разработке программ;
22. научиться выполнять объектно-ориентированный анализ задачи, выделять свойства и методы объектов;
23. научиться использовать инкапсуляцию для защиты данных объектов;
24. познакомится с понятиями «класс» и «абстрактный класс»;
25. познакомится с понятиями «инкапсуляция», «наследование», «полиморфизм»;
26. научиться проектировать несложные иерархии классов для прикладных задач;
27. познакомится с принципами разработки событийно-ориентированных программ;
28. научиться создавать программы с графическим интерфейсом на языках Python
29. научиться использовать готовые и создавать новые компоненты (виджеты) для сред быстрой разработки программ.

Тематическое планирование с указанием количества часов, отводимых на освоение каждой темы

11 класс

№ п/п	Тема занятия	Кол-во часов	Электронные цифровые образовательные ресурсы
1.	Простейшие программы. Диалоговые программы. Переменные. Консольный ввод и вывод данных. ИОТ №3-13-2022	1	
2.	Компьютерная графика. Процедуры	1	
3.	Обработка целых чисел. Обработка вещественных чисел	1	
4.	Случайные и псевдослучайные числа	1	
5.	Ветвления. Сложные условия	1	
6.	Циклы с условием	1	
7.	Анимация. Циклы по переменной	1	
8.	Циклы в компьютерной графике	1	
9.	Процедуры. Функции	1	
10.	Рекурсия	1	
11.	Символьные строки. Обработка символьных строк	1	
12.	Строки в функциях	1	
13.	Массивы. Ввод и вывод массивов	1	
14.	Суммирование элементов массива	1	
15.	Подсчёт элементов массива, удовлетворяющих условию	1	
16.	Поиск значения в массиве. Поиск максимального элемента в массиве	1	
17.	Матрицы. Сложность алгоритмов	1	
18.	Простые алгоритмы сортировки. Сортировка слиянием	1	
19.	Быстрая сортировка. Двоичный поиск	1	
20.	Обработка файлов	1	
21.	Целочисленные алгоритмы	1	
22.	Словари. Структуры	1	
23.	Стек, очередь, дек	1	
24.	Деревья. Графы	1	
25.	Динамическое программирование. Игровые модели	1	
26.	Что такое ООП?	1	
27.	Модель задачи: классы и объекты. Классы и объекты в программе	1	
28.	Скрытие внутреннего устройства	1	
29.	Иерархия классов. Классы-наследники	1	
30.	Событийно-ориентированное программирование. Обобщение и систематизация знаний	1	
31.	Годовая контрольная работа	1	
32.	Работа над ошибками годовой контрольной работы. Использование компонентов (виджетов)	1	

№ п/п	Тема занятия	Кол-во часов	Электронные цифровые образовательные ресурсы
33.	Создание компонентов. Модель и представление	1	
34.	Обобщение и систематизация знаний	1	