ИССЛЕДОВАНИЕ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ ИЗ КУРСА МАТЕМАТИКИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Информатика

Макаров П.В.

10 класс, Тверской лицей, г. Тверь

Научный руководитель: Наумова А.И., преподаватель информатики высшей категории, Тверской лицей, г. Тверь

Введение

В данной работе представлен материал по исследованию информационных моделей из курса математики на языке программирования Python (Пайтон).

Цель данной работы заключается в том, чтобы получить *дополнительные знания* по этой теме.

Задача состоит в том, чтобы подобрать *соответствующий материал* с последующей систематизацией, обобщением и иллюстрацией текста.

Работа состоит из двух частей: *описательной* (дана характеристика основных этапов разработки информационных моделей) и *практической* (приведён пример разработки алгоритма и программы (скрипта) исследования квадратного уравнения с последующей обработкой данных на компьютере).

Основные этапы разработки и исследования моделей на компьютере

Использование компьютера для исследования информационных моделей различных объектов и систем позволяет *изучить их изменения* в зависимости от значения тех или иных параметров. Процесс разработки моделей и их исследование на компьютере можно разделить на *несколько основных этапов*.

Описательная информационная модель. На первом этапе исследования объекта или процесса обычно строится *описательная информационная модель* на *естественном* языке. Такая модель выделяет *существенные*, с точки зрения целей проводимого исследования, свойства (параметры) объекта, а несущественными параметрами пренебрегает.

Формальная модель. На втором этапе создаётся *формальная модель*, т.е. описательная информационная модель записывается с помощью какого-либо

формального языка. В такой модели с помощью формул, уравнений, неравенств и т. д. фиксируются формальные соотношения между исходными и искомыми величинами, а также накладываются ограничения на допустимые значения этих величин.

Однако не всегда удаётся найти формулы, явно выражающие искомые величины через исходные данные. В таких случаях используются приближённые математические методы, позволяющие получать *результаты с заданной точностью*.

Компьютерная модель. На третьем этапе необходимо *формализованную информационную модель преобразовать в компьютерную модель*, т. е. выразить её на *понятном для компьютера языке*. Существуют два пути решения этой задачи:

- создание проекта на одном из языков программирования;
- построение компьютерной модели с использованием некоторого приложения, например электронных таблиц.

В процессе создания компьютерной модели полезно разработать *удобный графический интерфейс*, который позволяет *визуализировать* формальную модель, а также *реализовать интерактивный диалог* человека с компьютером на этапе исследования модели.

Компьютерный эксперимент. Четвёртый этап исследования информационной модели состоит в проведении *компьютерного эксперимента*. Если компьютерная модель существует в виде программы на одном из языков программирования, её нужно запустить на выполнение и получить результаты.

Если компьютерная модель исследуется в приложении, например в электронных таблицах, можно провести сортировку или поиск данных, построить диаграмму или график и т. д.

Анализ полученных результатов и корректировка исследуемой модели. Пятый этап представляет собой *анализ полученных результатов* и корректировку исследуемой модели. В случае отличия результатов, полученных при исследовании информационной модели, от измеренных параметров

реальных объектов можно сделать вывод, что на предыдущих этапах построения модели были допущены ошибки или неточности.

Например, при построении описательной качественной модели могут быть неправильно отобраны существенные свойства объектов, в процессе формализации могут быть допущены ошибки в формулах и т. д. В таких случаях необходимо провести корректировку модели, причём уточнение модели *может проводиться многократно, пока анализ результатов не покажет их соответствие изучаемому объекту*.

Визуализация формальных моделей. В процессе исследования формальных моделей часто производится их *визуализация*. Для визуализации алгоритмов используются *блок-схемы*, пространственных соотношений параметров объектов – *чертежи*, моделей электрических цепей – *электрические схемы*. При визуализации формальных моделей с помощью анимации может *отображаться динамика процесса*, *производится построение графиков изменения величин* и т. д.

В настоящее время широкое распространение получили *компьютерные интерактивные визуальные модели*. В таких моделях исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели [1].

Исследование квадратного уравнения на языке Python

Содержательная постановка задачи

Используя *компьютерную визуализацию*, рассмотрим решение квадратного уравнения $ax^2 + bx + c = 0$ в среде программирования Python, проведя *анализ всех возможных условий* получения корней (корня) с подключением модулей для *построения графика* приближённого решения задачи.

Формальная модель

В зависимости от коэффициентов квадратного уравнения его корни представляют собой числа *действительные* или *мнимые*, *различные* или *равные*, *положительные* или *отрицательные*. *Исследование уравнения состоит в*

установлении характера корней уравнения в зависимости от его коэффициентов.

От знака дискриминанта $d = b^2 - 4ac$ квадратного уравнения существенно зависит характер корней уравнения, так как он стоит перед знаком радикала в формуле:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

1. d > 0; корни уравнения *действительные, различные*. График квадратного уравнения пересекает ось x в двух точках.

   Полагая d > 0, исследуем *знак корней*. Для этого воспользуемся *формулами Виета* (сумма корней приведённого квадратного уравнения равна второму коэффициенту с противоположным знаком, а их произведение равно свободному члену). Будем для удобства считать, что a > 0.

   - d > 0, a > 0, c > 0. Корни *одного знака*, так как их произведение положительно. Если b < 0, то они *оба положительны*; если b > 0, то они *оба отрицательны*.
   - d > 0, a > 0, c < 0. Корни *разного знака*, один из них положителен, другой отрицателен.
   - d > 0, a > 0, c = 0. (уравнение “неполное” вида $ax^2 + bx = 0$). Один из корней равен нулю, знак другого противоположен знаку b.
2. d = 0; корни квадратного уравнения *действительные и совпадающие* (графически эта ситуация выражается в том, что парабола касается оси x), знак корней при a > 0 противоположен знаку b.
3. d < 0; корни *комплексно сопряжённые* (парабола не пересекает ось x) *(рис. 1)* [3].

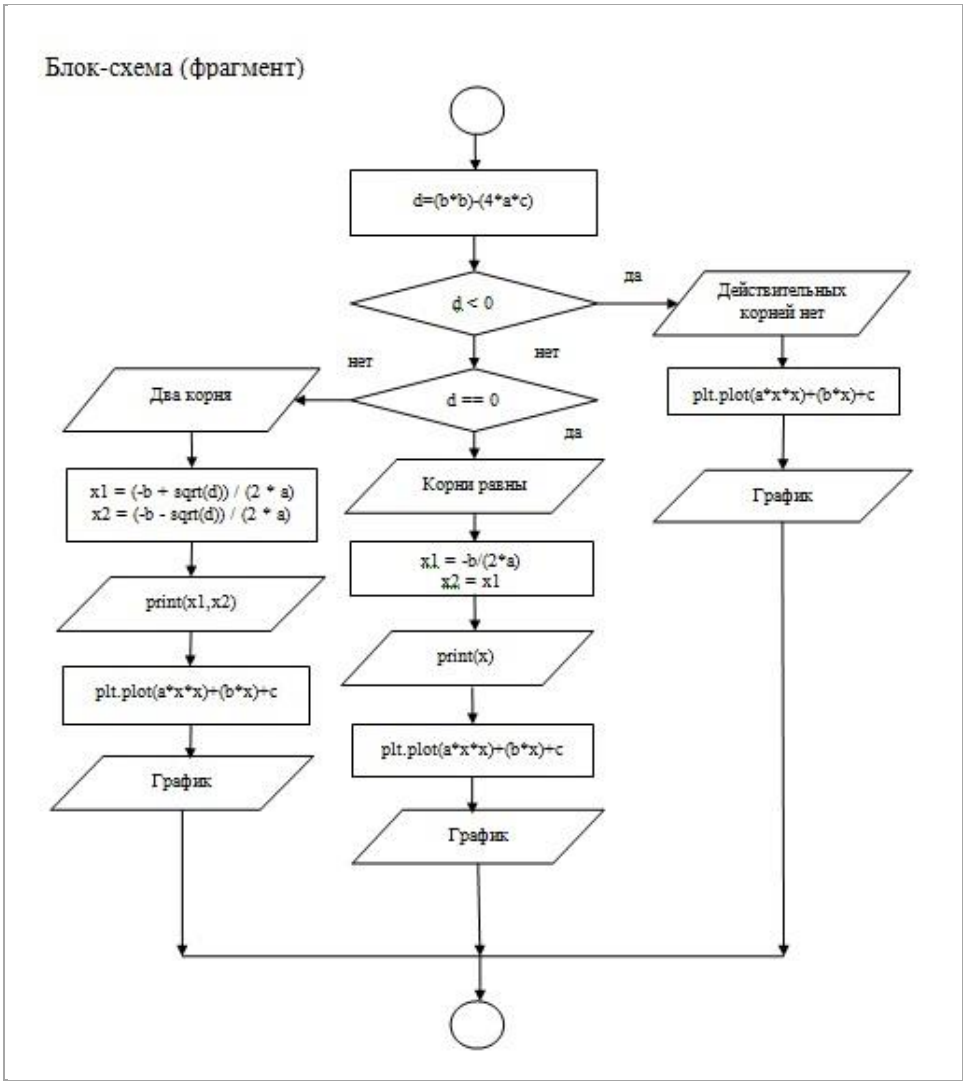


Рис. 1. Фрагмент блок-схемы алгоритма решения задачи

Компьютерная модель

На основании *формальной модели*, описывающей исследование квадратного уравнения, создадим *компьютерную модель* с использованием языка программирования Python (Пайтон), содержащую модули для работы с графикой, математические выражения и функции, несколько вложенных операторов if со сложными условиями, тип данных float (одинарная точность), формат выдачи данных, цикл for для проверки работы всех тестовых примеров (*табл. 1*), описание процедур пользователя [2,4]. Выполнение скрипта провести как непосредственно в среде программирования, так и с созданным выполняемым файлом .exe.

Программа (скрипт) с комментариями на языке Python

```
#полное исследование квадратного уравнения
```

```
#модули для работы с графикой
import numpy as np
import matplotlib.pyplot as plt
#подключить математическую функцию sqrt
from math import sqrt
#описание процедуры нахождения дискриминанта и корней
def Solution():
   d = (b * b) - (4 * a * c)          #найти дискриминант
   print("Дискриминант = {:5.3f}".format(d))
   if d > 0:          #условие для нахождения двух корней
      #найти значения двух корней
      x1 = (-b + sqrt (d)) / (2 * a)
      x2 = (-b - sqrt (d)) / (2 * a)
      print("Два корня:")
      #выдать на экран значения двух корней: x1 и x2
      print("x1 = {:5.3f}, x2 = {:5.3f}".format(x1, x2))
      Parabola(),Figure()
   else:
       if d == 0:   #условие для нахождения одного корня
          x1 = - b / (2 * a)
          x2 = x1
          print("Корни равны:")
          #выдать на экран значения равных корней: x1 и x2
          print("x1 = {:5.3f}, x2 = {:5.3f}".format(x1, x2))
          Parabola(),Figure()
       else:
          print("Действительных корней нет")
          Parabola(),Figure()
#описание процедуры нахождения вершины параболы
def Parabola():
   if a > 0:  #условие для полного квадратного уравнения
      print("Ветви параболы направлены вверх")
   else:
      print("Ветви параболы направлены вниз")
   #найти координаты вершины параболы: x0 и y0
   x0 = -b / (2 * a)
```

```
    y0 = (a * x0 * x0) + (b * x0) + c
    print("Координаты вершины параболы:")
    #выдать на экран координаты вершины параболы
    print("x0 = {:5.3f} y0 = {:5.3f}".format(x0, y0))
#описание процедуры построения графика
def Figure():
    plt.figure()        #создать график
    plt.xlim(-3, 3)     #установить пределы графика по оси x
    plt.ylim(-3, 3)     #установить пределы графика по оси y
    #установить имя графика
    plt.title("График квадратного уравнения")
    plt.xlabel("Ось x")#установить название оси x
    plt.ylabel("Ось y")#установить название оси y
    plt.grid()          #добавить сетку на график
    #Функция linspace() создает последовательность данных,
    #расположенных на числовой прямой в заданном интервале
    x = np.linspace(-3,3,100)
    #построить график сплошной линией синего цвета
    plt.plot(x,(a * x * x) + (b * x) + c ,"b-")
    plt.show()
#описание основной программы (скрипта)
print("Исследование квадратного уравнения"), print()
print("Введите количество проверочных тестов:")
n = int(input())
for i in range(n):
    print()
    print("Введите исходные данные уравнения через пробел:")
    #ввод с клавиатуры чисел типа float в одной строке
    a, b, c = map(float,input().split())
    if (a == 0) and (b == 0) and (c == 0):
         print("x - любое число")
    else:
        if (a == 0) and (b != 0):
            print("Линейное уравнение. Один корень")
            x = - c / b
            print("x = {:5.3f}".format(x))
```

```
        else:
            if (a == 0) and (b == 0) and (c!= 0):
               print("Неправильное уравнение")
            else:
                if (b == 0) or (c == 0):
                    print("Неполное квадратное уравнение")
                    Solution()
                else:
                    print("Полное квадратное уравнение")
                    Solution()
print()
print("Задержка экрана при выполнении файла .exe")
input("Для выхода из программы нажмите <Enter>")
print("Выполнение программы завершено")
```

Тестовые примеры

Таблица № 1. Тестовые примеры

№ теста	Проверка условия	Исходные данные			Результат
		a	b	c	
1.	a = 0, b = 0, c = 0	0	0	0	Все коэффициенты равны нулю. x – любое число.
2.	a = 0, b <> 0	0	2	1	Линейное уравнение. Один корень: x = -0.500
3.	a = 0, b = 0, c <> 0	0	0	2	Неправильное уравнение.
d > 0					
4.	d > 0 a > 0, c > 0 b < 0	1	-3	2	Корни оба положительны: x_1 = 2.000; x_2 = 1.000. Ветви параболы направлены вверх. Парабола пересекает ось x в двух местах.
5.	d > 0 a < 0, c < 0 b < 0	-1	-3	-2	Корни оба отрицательны: x_1 = -2.000; x_2 = -1.000. Ветви параболы направлены вниз. Парабола пересекает ось x в двух местах.

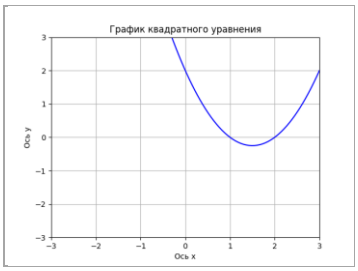
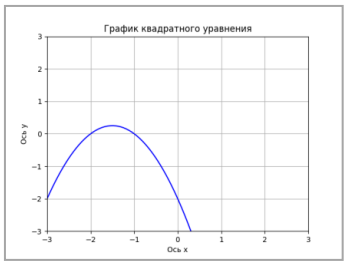
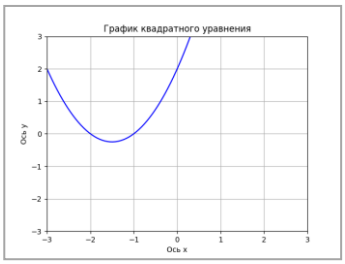
6.	$d > 0$ $a > 0, c > 0$ $b > 0$	1	3	2	Корни оба отрицательны: $x_1 = -1.000$; $x_2 = -2.000$. Ветви параболы направлены вверх. Парабола пересекает ось x в двух местах.
7.	$d > 0$ $a < 0, c < 0$ $b > 0$	-1	3	-2	Корни оба положительны: $x_1 = 1.000$; $x_2 = 2.000$. Ветви параболы направлены вниз. Парабола пересекает ось x в двух местах.
8.	$d > 0$ $a > 0, b > 0$ $c < 0$	1	1	-2	Корни разного знака: $x_1 = 1.000$; $x_2 = -2.000$. Ветви параболы направлены вверх. Парабола пересекает ось x в двух местах.
9.	$d > 0$ $a < 0, b > 0$ $c > 0$	-1	1	2	Корни разного знака: $x_1 = -1.000$; $x_2 = 2.000$. Ветви параболы направлены вниз. Парабола пересекает ось x в двух местах.
10.	$d > 0$ $a > 0, c = 0$	1	1	0	Неполное уравнение. Один из корней равен нулю, знак другого противоположен знаку b: $x_1 = 0.000$; $x_2 = -1.000$. Ветви параболы направлены вверх. Парабола пересекает ось x в двух местах.
11.	$d > 0$ $a < 0, b = 0$	-1	0	1	Неполное уравнение. Корни разного знака: $x_1 = -1.000$; $x_2 = 1.000$. Ветви параболы направлены вниз. Парабола пересекает ось x в двух местах.
d = 0					
12.	$d = 0$ $a > 0, c > 0$ $b > 0$	1	2	1	Действительные совпадающие отрицательные корни. Знак корней при $a > 0$ противоположен знаку b: $x_1 = -1.000$; $x_2 = -1.000$. Ветви параболы направлены вверх. Парабола касается оси x.
13.	$d = 0$ $a < 0, c < 0$	-1	2	-1	Действительные совпадающие положительные корни. Знак корней при $a < 0$ равен знаку b:

	b > 0				x_1 = 1.000; x_2 = 1.000. Ветви параболы направлены вниз. Парабола касается оси x.
d < 0					
14.	d < 0 a > 0, c > 0 b > 0	2	1	2	Полное квадратное уравнение. Действительных корней нет. Корни комплексно сопряжённые. Ветви параболы направлены вверх. Парабола не пересекает ось x.
15.	d < 0 a < 0, c < 0 b > 0	-2	1	-2	Полное квадратное уравнение. Действительных корней нет. Корни комплексно сопряжённые. Ветви параболы направлены вниз. Парабола не пересекает ось x.

Компьютерный эксперимент

1. Для работы с графикой подключить библиотеку matplotlib: в командной строке набрать: *pip install matplotlib*.
2. Войти в программную среду *Python* (*Пайтон) – IDLE*; выполнить команду [*File – New File*]; для повторного выполнения – [*File – Open*].
3. В текстовом редакторе набрать программу (скрипт) с пояснениями – комментариями на русском языке.
4. Запустить программу на выполнение, выполнив команду: [*Run – Run Module (F5)*].
5. В появившемся окне нажать на <*OK*>, далее сохранить файл, указав *имя* программы и *место* сохранения.
6. В режиме выполнения ввести *исходные данные* по подготовленным тестам и получить *соответствующие результаты (табл. 2)*.
7. При каждом выполнении цикла высвечивается окно *с построенным графиком*; для его сохранения выполнить команду, нажав на кнопку внизу графика (справа) – [*Save the figure*].
8. После проверки *всех подготовленных тестов* выйти из программы, выполнив команду [*File – Exit*]; при завершении выполнения файла .exe – нажать на <*Enter*>.

Таблица № 2. Построение графиков по тестовым примерам № 4 - № 6 **(d > 0)**

Результаты, полученные при выполнении скрипта на компьютере		
Тест № 4	**Тест № 5**	**Тест № 6**
		
Оба корня положительны. Ветви параболы направлены вверх. Парабола пересекает ось x в двух местах.	Оба корня отрицательны. Ветви параболы направлены вниз. Парабола пересекает ось x в двух местах.	Оба корня отрицательны. Ветви параболы направлены вверх. Парабола пересекает ось x в двух местах.

Анализ полученных результатов

В зависимости от *начальных значений* коэффициентов, *свободного члена* квадратного уравнения и *дискриминанта* (*табл. 1)* получены *результаты*, которые *полностью соответствуют результатам всех тестовых примеров*.

Заключение

В данной работе показано решение математической задачи на современном высокоуровневом профессиональном языке программирования Python: проведено *полное исследование квадратного уравнения* с использованием вложенных сложных условий, подключения модулей для работы с графикой и созданием функций пользователя. Показаны примеры выполнения программы (скрипта). Проведён *сравнительный анализ планируемых и полученных результатов* компьютерного эксперимента.

В ходе разработки проекта получены дополнительные знания по двум дисциплинам: *математике* и *информатике*. Проведённый компьютерный эксперимент наглядно показывает *практическую значимость* проведённых исследований.

Python (Пайтон) – *современный развивающийся язык*, изучение которого начинается в *профильных классах* общеобразовательных учреждений на примере *использования его основных конструкций*, что достаточно *наглядно* показано в данной работе.

Программное обеспечение

1. Операционная система Windows 10
2. Среда программирования Python – 3.8.0
3. Приложение Microsoft Office Word 2010

Список использованных источников и литературы

1. Информатика и ИКТ. Профильный уровень : учебник для 11 класса / Н. Д. Угринович. – 2-е изд., испр. и доп. – М. : БИНОМ. Лаборатория знаний, 2009. – 308 с. : ил.
2. Информатика: Учеб. пособие для 10-11 кл. общеобразоват. учреждений / Л.З. Шауцукова. – М.: Просвещение, 2000. – 416 с.: ил.
3. Исследование квадратного уравнения [Электронный ресурс]. – Режим доступа: https://scask.ru/f_book_el_math.php?id=62.
4. Создание графического интерфейса на Python [Электронный ресурс]. – Режим доступа: https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter.