

Рочева Марина Геннадьевна, старший преподаватель кафедры высшей математики, ФГБОУ ВО «Ухтинский государственных технических университет», г. Ухта

Гамова Ульяна Николаевна, студент факультета экономики, управления и информационных технологий, ФГБОУ ВО «Ухтинский государственных технических университет», г. Ухта

Тодоров Алексей Сергеевич, студент факультета экономики, управления и информационных технологий, ФГБОУ ВО «Ухтинский государственных технических университет», г. Ухта

АВТОМАТИЗАЦИЯ ПРОЦЕССА ИЗВЛЕЧЕНИЯ КВАДРАТНОГО КОРНЯ И ЕЕ ПРИМЕНЕНИЕ ПРИ РЕШЕНИИ КВАДРАТНЫХ УРАВНЕНИЙ

Аннотация: Данная работа посвящена разработке программы по вычислению квадратного корня из любого действительного числа. Данная тема актуальна в студенческой среде, так как операция извлечения квадратного корня из отрицательного числа не реализуется на калькуляторе. Рассмотрены основные понятия и приложения теории комплексных чисел, изложен процесс выбора языка программирования для реализации программы, отладка и тестирование программы.

Ключевые слова: квадратный корень из отрицательного числа, комплексное число, квадратное уравнение, дискриминант, создание программы.

Annotation: This work is devoted to the development of a program for calculating the square root of any real number. This topic is relevant in the student environment, since the operation of extracting the square root from a negative

number is not implemented on a calculator. The basic concepts and applications of the theory of complex numbers are considered, the process of choosing a programming language for implementing a program, debugging and testing a program are outlined.

Keywords: square root of a negative number, complex number, quadratic equation, discriminant, program creation.

Операция извлечения квадратного корня из действительного числа в школе и в вузе рассматривается по-разному. В школьном курсе математики говорится, что действительного корня из отрицательного числа не существует. А в случае возникновения квадратного корня, подкоренное выражение которого отрицательно, в ответе записывали «решений не существует». Школьники сталкиваются с данной ситуацией, например, при решении квадратных уравнений с отрицательным дискриминантом.

Однако, уже на первых занятиях по математике в техническом вузе становится известно, что данное значение существует, но является не действительным, а комплексным числом. Комплексным числом z называется выражение вида $z = x + iy$, где $x \in R$ и $y \in R$, а i – мнимая единица. Основное свойство числа i в том, что $i^2 = -1$ (или $\sqrt{-1} = i$). Число x называется действительной частью комплексного числа z и обозначается $x = Re z$ (от франц. *reele* – «действительный»), а y – мнимой частью z , $y = Im z$ (от франц. *imaginaire* – «мнимый»). Множество комплексных чисел обозначается латинской буквой C [1].

Необходимость введения комплексных чисел возникла в XVI веке в связи с изучением кубических уравнений, когда оказалось необходимым извлекать квадратные корни из отрицательных чисел.

Впервые мнимые величины появились в известном труде «Великое искусство, или об алгебраических правилах» Дж. Кардано (1545 г.), который счёл их непригодными к употреблению. Символ $\sqrt{-1} = i$ предложил Эйлер (1777 г., опубл. 1794 г.), взявший для этого первую букву слова «*imaginaris*» (лат. «воображаемый»).

Комплексные числа крепко укоренились в методах исследования окружающего мира – как в математике, так и в физике и других областях точных наук. Комплексные числа используются для облегчения вычислений, но в качестве конечных данных всё равно используется только вещественная часть комплексного числа, так как мнимую часть отбрасывают из-за того, что её нельзя представить.

Актуальность данной работы вытекает из того, что возможность извлечения корня из любого действительного числа на калькуляторе ограничена: техника не вычисляет квадратный корень из отрицательного числа. Целью нашей работы является создание программного продукта по извлечению квадратного корня из любого действительного числа. Ведь действительно, это очень упрощает математические вычисления. Для достижения поставленной цели были решены следующие задачи: изучить историю возникновения комплексных чисел, основные понятия и приложения теории комплексных чисел, выбор языка программирования, отладка и тестирование готового продукта.

Перейдем к описанию экспериментальной части. Первоначально код программы по вычислению квадратного корня из любого действительного числа был написан на языке программирования Pascal ABC (Рисунок 1), но интерфейс программы был не очень удобным.

```

var a,x: real;
begin
  writeln ( 'Введите число:  ' );
  readln ( a );
  if a<0 then
  begin
    x:= sqrt ( abs ( a ) );
    writeln ( 'X= ', '- ', x : 1 : 2, 'i ' );
  end;
  if a>=0 then
  begin
    x:= sqrt ( a );
    writeln ( 'X= ', x : 1 : 2);
  end;
end.

```

Рисунок 1 – Код программы на языке Pascal ABC

Следующим этапом стало написание кода программы в среде программирования Excel VBA (Рисунок 2 Рисунок 2 – Код программы в среде программирования Excel VBA) с удобным интерфейсом. VBA (Visual Basic for Applications) — это упрощенная версия Visual Basic, встроенная в множество продуктов линейки Microsoft Office.

```

If D.Value >= 0 Then
  X.Value = ((Val(D.Text) ^ 0.5))
End If
If D.Value < 0 Then
  X.Value = "-" + CStr(((Abs(Val(D.Text))) ^ 0.5)) + "j"
End If

```

Рисунок 2 – Код программы в среде программирования Excel VBA

И, наконец, завершающим этапом для реализации задуманного алгоритма стала среда быстрой разработки C++ Builder, в которой в качестве языка программирования используется язык C++ Builder (C++ Builder Language) (Рисунок 3 – Код программы на языке программирования C++ Builder). Язык C++ Builder – это расширенный C++, в котором, например, есть строковый (AnsiString) и логический (bool) типы, которых нет в классическом C++ [2].

```

float a;
float x;
a = StrToInt(Edit1->Text);
StringGrid1 -> Cells[0][0] = "Корень";
StringGrid1 -> Cells[0][1] = "x =";
//нахождение корня, если число больше или равно 0
if (a >= 0) {
    x= (sqrt(a));
    StringGrid1 -> Cells[1][1] = x;
}
//нахождение корня,если число меньше 0
if (a < 0) {
    x = (sqrt(-1*a));
    String z1_str = "-" +FloatToStr(x)+"i";
    StringGrid1 -> Cells[1][1] = z1_str;
}

```

Рисунок 3 – Код программы на языке программирования C++ Builder

С операцией извлечения квадратного корня студенты встречаются при решении квадратных уравнений в теории комплексных чисел, а так же в ходе решения линейных однородных дифференциальных уравнений второго порядка с постоянными коэффициентами и линейных неоднородных дифференциальных уравнений с правой частью специального вида. Поэтому авторы статьи своей целью поставили создание программы по решению квадратных уравнений.

Поэкспериментировав с языками программирования ранее, для написания кода программы по решению квадратных уравнений был выбран C++ Builder. Код программы представлен ниже.

```

#include <vcl.h>
#include <math.h>
#pragma hdrstop
#include "Statia.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
//создание формы интерфейса
//-----

```

```

__fastcall TForm4::TForm4(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
//кнопка "рассчитать"
void __fastcall TForm4::Button1Click(TObject *Sender)
{
float a;      //присвоение переменным типа данных float
float b;
float c;
float D,x1,x2;
a = StrToInt(Edit1->Text); //введение значений в поле
b = StrToInt(Edit2->Text);
c = StrToInt(Edit3->Text);
StringGrid1 ->Cells[0][0] = "Корни";      //название главного столбца таблицы
StringGrid1 -> Cells[0][1] = "x1 =";      //названия строк
StringGrid1 -> Cells[0][2] = "x2 =";

D = b*b - 4*a*c;      //расчёт дискриминанта
//нахождение корней, если дискриминант больше 0
if (D>0) {
    x1 = ((b*(-1)-sqrt(D))/(2*a));      //вычисление корней
    x2 = ((b*(-1)+sqrt(D))/(2*a));
    StringGrid1 -> Cells[1][1] = x1;      //вывод значений в поля
    StringGrid1 -> Cells[1][2] = x2;
}
//нахождение корней, если дискриминант равен 0
if (D == 0) {
    x1 = ((-1)*b)/(2*a);      //вычисление корней
    x2 = ((-1)*b)/(2*a);
    StringGrid1 -> Cells[1][1] = x1;      //вывод значений в поля
    StringGrid1 -> Cells[1][2] = x2;
}
//нахождение корней, если дискриминант меньше 0

```

```

if (D<0) {
    //вычисление корней с комплексным числом
    String x1_str = FloatToStr(b*(-1.0)/(2*a))+ "-" + FloatToStr(sqrt(-1*D)/(2.0*a))+"i";
    String x2_str = FloatToStr(b*(-1.0)/(2*a))+ "+" + FloatToStr(sqrt(-1*D)/(2.0*a))+"i";
    StringGrid1->Cells[1][1] = x1_str; //вывод значений в поля
    StringGrid1 -> Cells[1][2] = x2_str;
}
}
//-----
//кнопка "очистить"
void __fastcall TForm4::Button2Click(TObject *Sender)
{
    Edit1->Clear();. //очистка полей
    Edit2->Clear();
    Edit3->Clear();
    for (int i=0; i < StringGrid1->RowCount; i++) { //очистка таблицы
        StringGrid1->Cols[i]->Clear();
    }
}
//-----

```

Программа имеет удобный интерфейс, изображенный на рисунке 4.

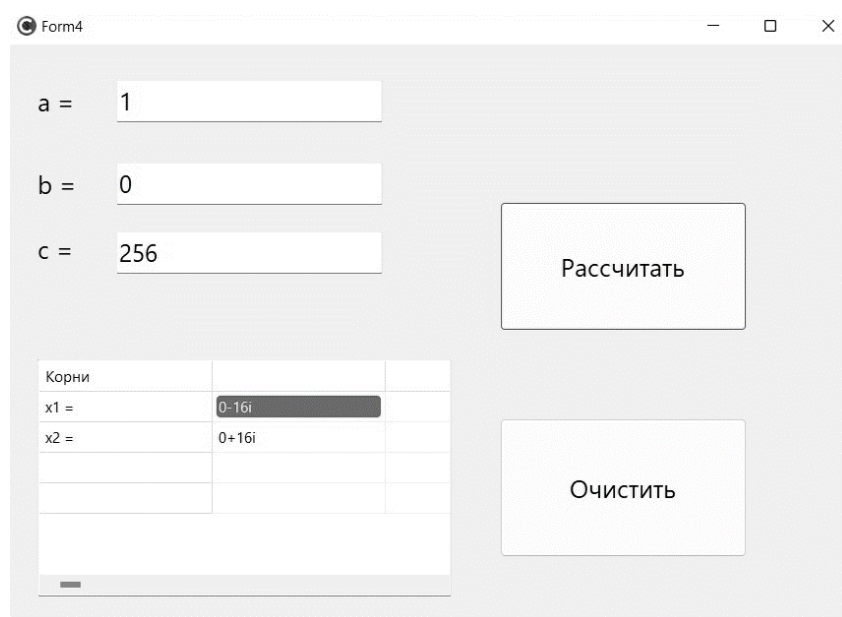


Рисунок 4 – Интерфейс программы

В первые три поля вводятся значения коэффициентов квадратного уравнения. Кнопка «Рассчитать» запускает процесс решения (вычисляется дискриминант и корни уравнения) и выводит значения двух корней на экран. Кнопка «Очистить» стирает значения из всех полей.

На рисунке 4 показано, как работает программа при решении квадратного уравнения $x^2 + 256 = 0$, дискриминант которого отрицательный ($D < 0$). В ответе получены два комплексно-сопряженных корня. Так же была проведено тестирование программы в случае $D > 0$ (два различных действительных корня) и $D = 0$ (два совпадающих действительных корня).

Данная программа была апробирована студентами группы ИВТ-21о-Б Ухтинского государственного технического университета на занятиях по математике при изучении темы «Линейные однородные дифференциальные уравнения с постоянными коэффициентами» в качестве контролирующего средства вычислительных операций, сделанных студентами на бумаге.

Подведем итог. Так как операция извлечения квадратного корня из отрицательного числа практически не автоматизирована, то сделанная авторами статьи работа будет полезна учащимся вузов. Разработанная программа поможет студентам как при проведении сложных расчетов, так и при проверке правильности решений на бумаге. В будущем планируется создание других алгоритмов и программ, востребованных студентами в процессе обучения.

Библиографический список:

1. Тренировочные задачи и упражнения по математике для студентов технических вузов [Текст]: учеб. пособие / О. А. Сотникова, М. Г. Рочева, М. С. Хозяинова. – 2-е изд., испр. и доп. – Ухта: УГТУ, 2017.
2. Технология программирования для студентов высших учебных заведений [Текст]: учеб. пособие/ Г. Н. Гатин. – 2-е изд., перераб. - Ухта: УГТУ, 2012.