

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Информационных систем

ОТЧЕТ
по практической работе №1
по дисциплине «Объектно-ориентированное программирование»

Студент гр. 8374	_____	Пихтовников К.С.
Студент гр. 8374	_____	Подсекин Г.С.
Преподаватель	_____	Егоров С.С.

Санкт-Петербург

2021

Задание на практическую работу

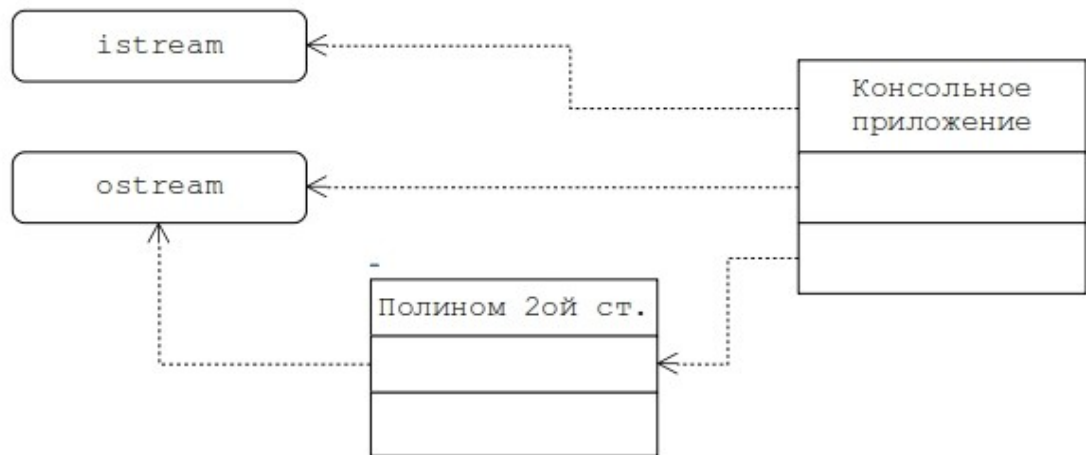


Рис.1. Диаграмма классов работы №1

Создать консольное приложение согласно представленной на рис.1 диаграмме классов, предназначенное для вычисления корней полинома 2-ой степени $p(x) = a \cdot x^2 + b \cdot x + c$ ($a \neq 0$) и его значения для заданного аргумента x на **множестве целых чисел**.

Приложение должно включать основной модуль (функция `main`), модуль «`application`» и модуль «`polinom`».

В **основном модуле** консольного приложения (для языка C++ — это модуль с функцией `main`) должен создаваться объект класса "Консольное приложение" и вызываться его метод, который предоставляет пользователю **меню команд** приложения.

Модуль «**application**» должен содержать спецификацию класса "Консольное приложение" и реализацию его методов. Один из методов должен реализовывать меню команд приложения, включающее:

- команду, инициирующую ввод коэффициентов a , b , c (до ввода должны быть заданы значения по умолчанию);
- команду, инициирующую расчета корней полинома и вывод результатов расчета;
- команду, инициирующую ввод значения аргумента x (по умолчанию равен 0), расчет значения и его вывод;
- команду, инициирующую вывод текстового представления полинома в указанной форме $p(x)$;
- команду, инициирующую вывод текстового представления полинома в канонической форме;

- команду выхода из приложения.

Модуль «**polinom**» должен содержать спецификацию класса "Полином 2ой степени" и реализацию его методов, необходимых для реализации цели разрабатываемого приложения. Описание класса должно использовать вместо типа `double` (вещественное число, заданное в условии) абстрактный тип ***number***, описание которого должно задаваться в отдельном заголовочном файле `number.h` с помощью оператора **`typedef int number`** (для C++).

Требуется реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и протестировать на них программу. Оформить отчет, сделать выводы по работе.

Спецификация классов

Класс Tapplication

Предназначен для выполнения функций ввода коэффициентов полинома, значения аргумента, инициализации процесса вычисления корней, инициализации процесса вычисления и вывода полинома в классической и канонической формах.

Метод/атрибут	Описание
Метод exes()	Формальных параметров нет, тип void, область видимости-public. В этом методе идет вызов функции menu(), задается конкретное действие, которое пожелал сделать пользователь и результат выводится на экран.
Метод menu()	Формальных параметров нет, тип возвращаемого значения-int, область видимости-private. Вывод на экран необходимого меню, с помощью которого пользователь взаимодействует с программой.
Метод Tapplication()	Конструктор класса

Таблица 1. Класс Tapplication

Класс Tpolinom

Методы и атрибуты данного класса необходимы для выполнения цели разрабатываемой программы (например, получение значений коэффициентов полинома, вычисление дискриминанта, вывод уравнения на экран).

Метод/атрибут	Описание
Атрибут number a, b, c	область видимости – private, хранит значение коэффициентов. По умолчанию коэффициенты равны: a=1, b=2, c=1.
Атрибут EPrintMode printMode	область видимости – private, содержит вид уравнения для печати, который выбрал пользователь
Метод TPolinom(number,number,number)	Конструктор класса
Метод Void setPrintMethod(EPrintMode ePrintMethod)	Тип формального параметра - EPrintMode, область видимости public. Метод устанавливает вид полинома, в котором его необходимо вывести (классический или канонический)
Методы getA(), getB(), getC()	Формальных параметров нет, тип number, область видимости public.

	Предназначены для получения коэффициентов a, b, c.
Метод <code>Int QuantityOfRoots()</code>	Формальных параметров нет, тип возвращаемого значения - <code>int</code> , область видимости <code>public</code> . Возвращает количество корней полинома
Метод <code>number getValue(number x)</code>	Тип формального параметра - <code>number</code> , тип возвращаемого значения - <code>number</code> , область видимости <code>public</code> . Метод вычисляет и возвращает значение полинома.
Метод <code>number Discriminant()</code>	Формальных параметров нет, тип возвращаемого значения - <code>number</code> , область видимости <code>private</code> . Возвращает значение дискриминанта
Метод <code>Bool RootsInteger (number*ArrayOfRoots, int quantityRoots, number a, number b, number c)</code>	Тип возвращаемого значения - <code>bool</code> , область видимости <code>private</code> . Типы формальных параметров: указатель на массив с корнями (<code>number*</code>), <code>int</code> количество корней, <code>number</code> коэффициенты a, b, c. Метод позволяет определить, являются ли корни уравнения целыми числами.
Метод <code>number *Roots()</code>	Формальных параметров нет, тип возвращаемого значения - <code>number</code> , область видимости <code>public</code> . Возвращает указатель на массив с корнями.
Метод <code>ostream& operator << (ostream& os, TPolinom& p)</code>	Тип возвращаемого значения – <code>ostream object</code> , область видимости – <code>public</code> . Данный метод выводит на экран полином в классической или канонической форме.

Таблица 2. Класс *Tpolinom*

Диаграмма классов

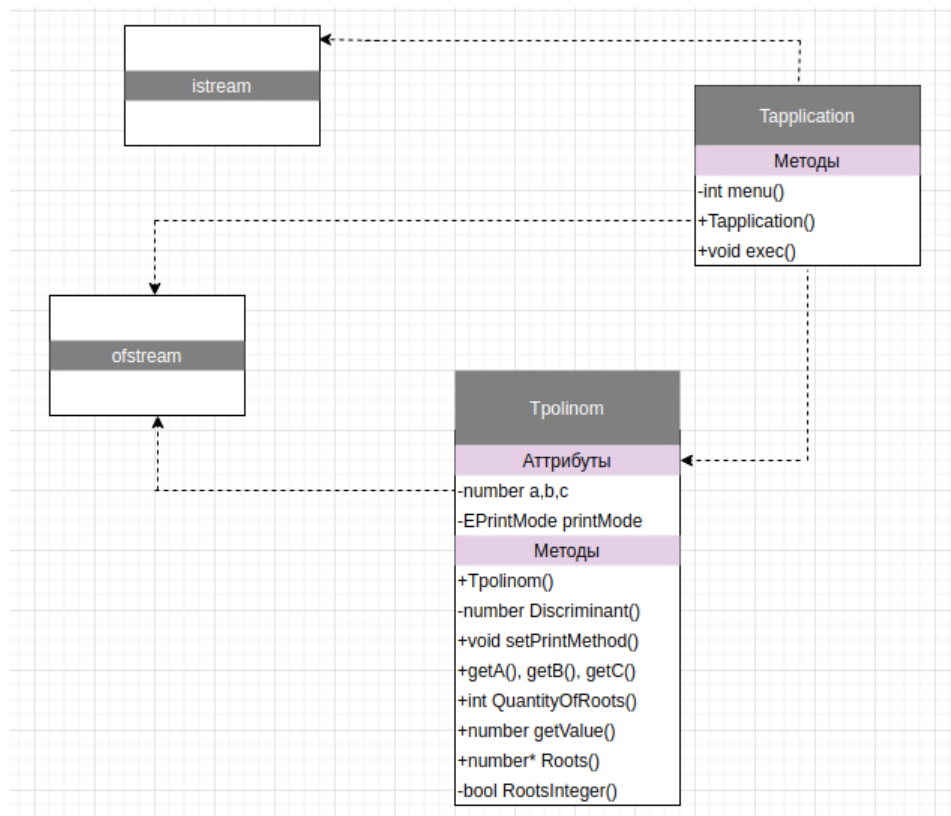


Рис.2. Реализация диаграммы классов работы №1

Символ	Значение
+	public - открытый доступ
-	private - только из операций того же класса
#	protected - только из операций этого же класса и классов, создаваемых на его основе

Таблица 3. Обозначение атрибутов и методов класса

Описание контрольного примера с исходными и ожидаемыми (расчетными) данными

Пример 1:

Исходные данные:

Коэффициенты:

$$a=1, b=3, c=-4$$

Ожидаемые данные:

$$x_1=-4, x_2=1$$

$$p(2)=2*2+3*2-4=6$$

Классический вид: x^2+3x-4

Канонический вид: $1*(x-1)*(x+4)$

Пример 2:

Исходные данные:

Коэффициенты:

$$a=1, b=6, c=9$$

Ожидаемые данные:

$$x_1=x_2=-3$$

$$p(3)=3*3+6*3+9=36$$

Классический вид: x^2+6x+9

Канонический вид: $1*(x+3)^2$

Пример 3:

Исходные данные:

Коэффициенты:

$$a=-1, b=4, c=0$$

Ожидаемые данные:

$$x_1=4, x_2=0$$

$$p(1)=-1*1*1+4*1=3$$

Классический вид: $-1x^2+4x$

Канонический вид: $-1*x*(x-4)$

Пример 4:

Исходные данные:

Коэффициенты:

$$a=2, b=4, c=7$$

Ожидаемые данные:

Корней нет

$$p(5)=2*5*5+4*5+7=77$$

Классический вид: $2x^2+4x+7$

Канонический вид: полином не имеет корней, поэтому его невозможно вывести в канонической форме.

Скриншоты программы на контрольных примерах

Пример 1:

```
Enter a,b,c:
>1 3 -4
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>3
There are two roots: x1=1      x2=-4
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>2
P(2)=6
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
1x^2+3x-4
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
1*(x-1)*(x+4)
```

Пример 2:

```
Enter a,b,c:
>1 6 9
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>3
There is one root: x=-3
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>3
P(3)=36
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
1x^2+6x+9
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
1*(x+3)^2
```

Пример 3:

```
Enter a,b,c:
>-1 4 0
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>3
There are two roots: x1=0      x2=4
```

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>1
P(1)=3
```

```

1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
-1x^2+4x

```

```

1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
-1*x*(x-4)

```

Пример 4:

```

Enter a,b,c:
>2 4 7
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>3
No roots on the field of integers

```

```

1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>5
P(5)=77

```

```

1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
2x^2+4x+7

```

```

1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
The polynomial has no roots, so it is impossible to derive in canonical form

```

Вывод

В ходе данной лабораторной работы было создано консольное приложение согласно представленной на рис.1 диаграмме классов. Приложение включает в себя основной модуль (функция main), в котором создается объект класса "Консольное приложение" и вызывается его метод, предоставляющий пользователю меню команд приложения; модуль «application», содержащий спецификацию класса "Консольное приложение" и реализацию его методов; модуль «polinom», содержащий спецификацию класса "Полином 2ой степени" и реализацию его методов, необходимых для реализации цели разрабатываемого приложения.

Помимо этого, была создана диаграмма классов (рис.2) и произведена отладка работы программы. Разработаны контрольные примеры с исходными и ожидаемыми данными, которые затем были протестированы в созданном консольном приложении. Все результаты совпали.