

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**  
**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**  
**Дисциплина: Платформо-независимое программирование**

Работу выполнил: \_\_\_\_\_ А. А. Костров

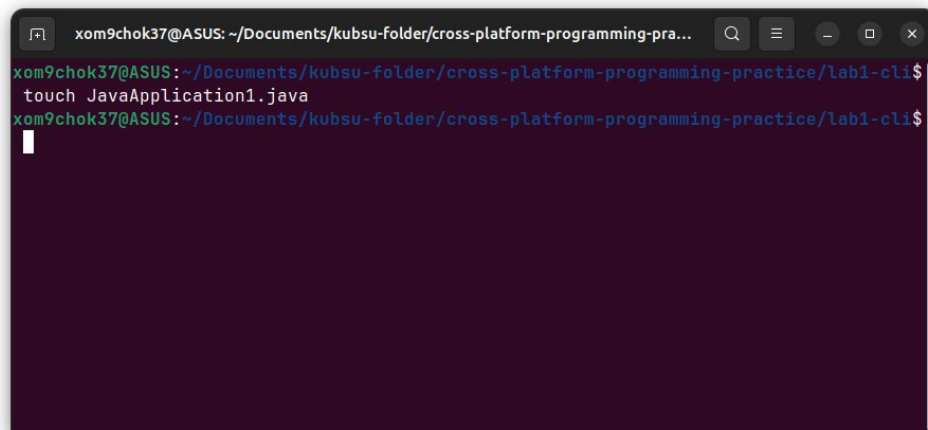
Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_ В. И. Шиян

*Тема.* Работа в командной строке, компиляция и запуск на выполнение.

*Цель.* Освоить основы работы в командной строке Java. Изучить структуру и синтаксис простой программы. Посмотреть возможности языка.

1. Напишем, скомпилируем и выполним программу *Калькулятор* и примеры из командной строки (рисунки 1-14).

A terminal window with a dark background. The prompt is 'xom9chok37@ASUS: ~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli\$'. The user enters 'touch JavaApplication1.java' and presses enter. The prompt returns.

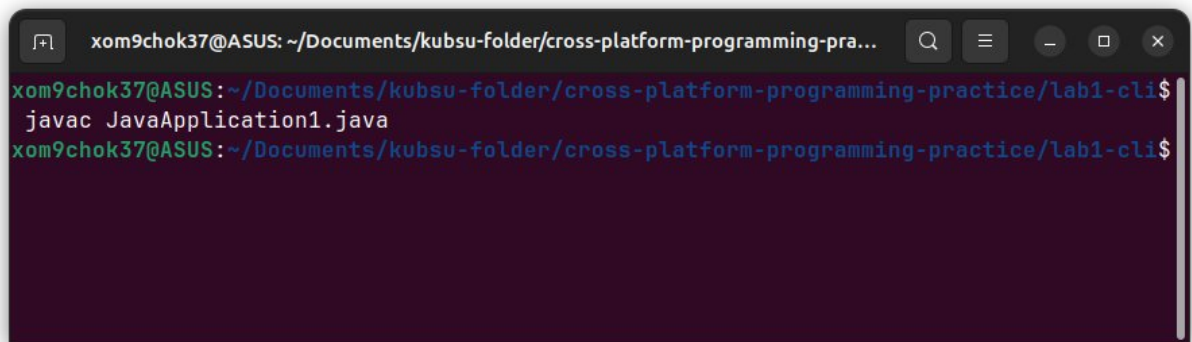
```
xom9chok37@ASUS: ~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$  
touch JavaApplication1.java  
xom9chok37@ASUS: ~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$
```

Рисунок 1 — Создадим файл *JavaApplication1.java*

A code editor window titled 'JavaApplication1.java' showing the source code of a Java program. The code defines a main class, an inner Adder class, and a Calculator class.

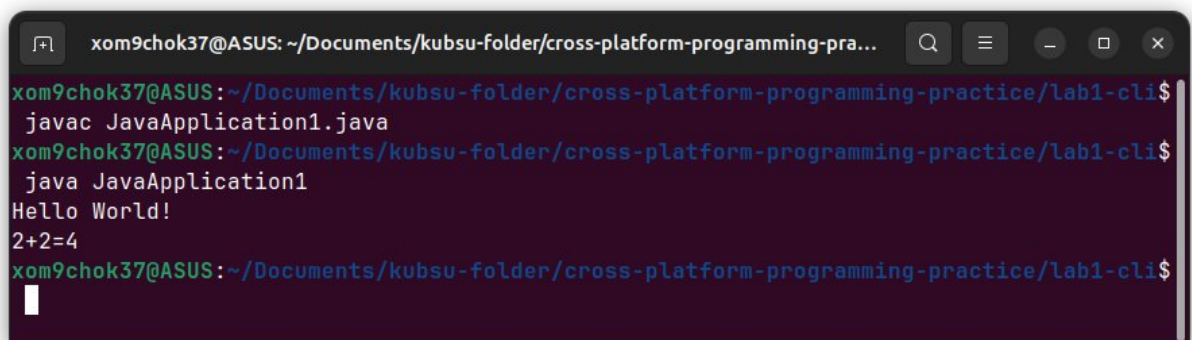
```
public class JavaApplication1 {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
        Calculator calc=new Calculator();  
        System.out.println("2+2="+calc.sum(2,2));  
    }  
  
    public static class Adder {  
        private int sum;  
        public Adder() { sum=0; }  
        public Adder(int a) { this.sum=a; }  
        public void add(int b) { sum+=b; }  
        public int getSum() { return sum; }  
    }  
  
    public static class Calculator {  
        public int sum(int... a) {  
            Adder adder=new Adder();  
            for (int i:a) {  
                adder.add(i);  
            }  
            return adder.getSum();  
        }  
    }  
}
```

Рисунок 2 — Напишем код программы *Калькулятор*



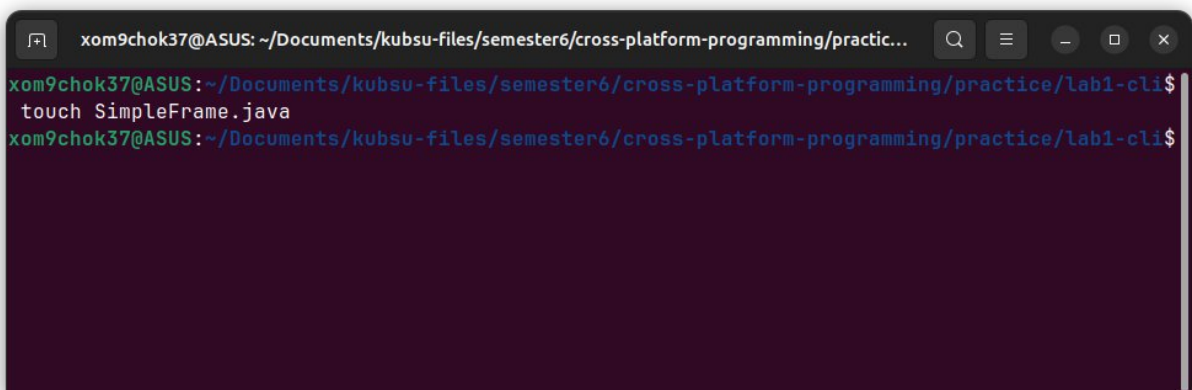
```
xom9chok37@ASUS: ~/Documents/kubsu-folder/cross-platform-programming-pra...
xom9chok37@ASUS:~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$
javac JavaApplication1.java
xom9chok37@ASUS:~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$
```

Рисунок 3 — Скомпилируем программу



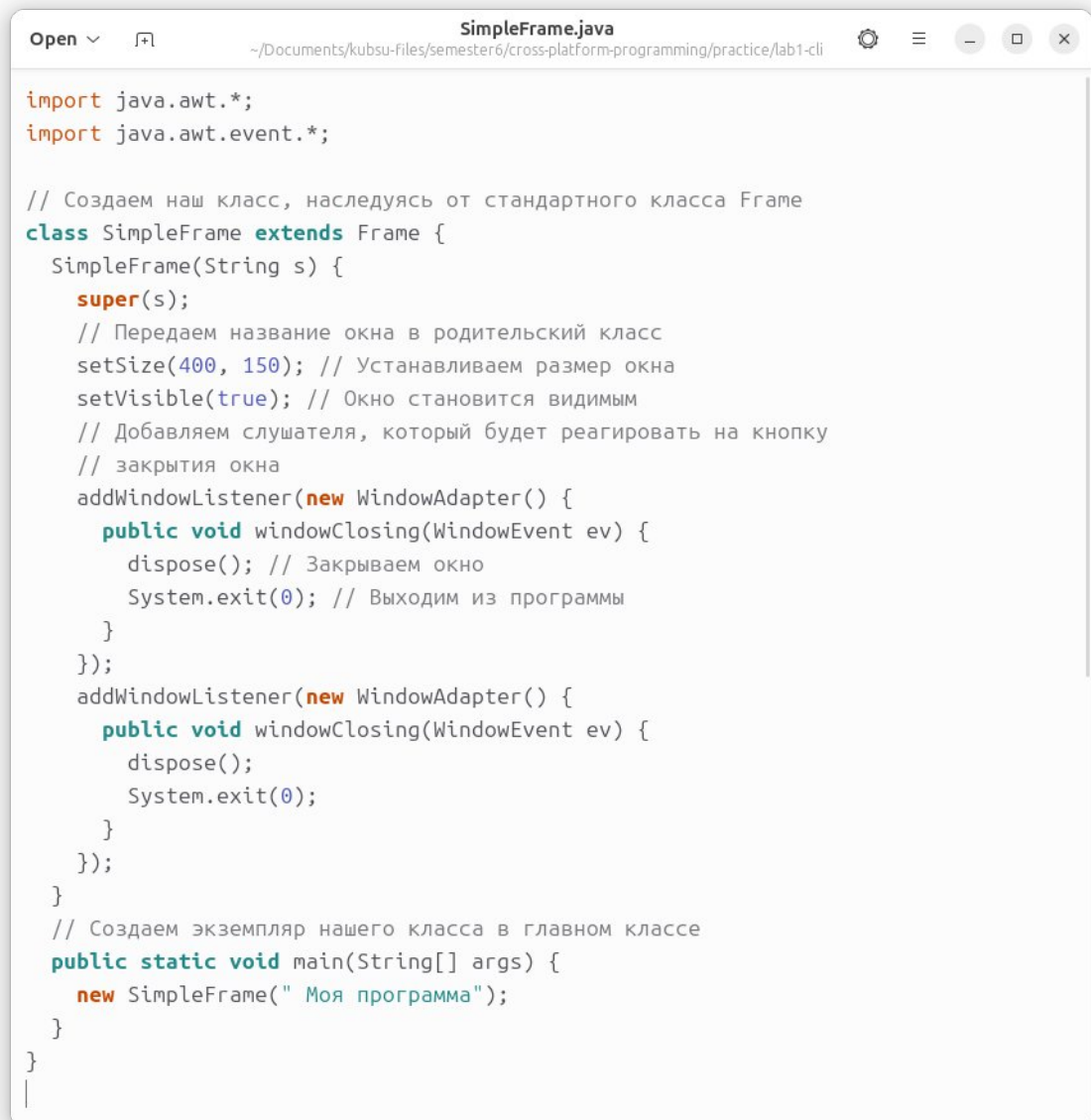
```
xom9chok37@ASUS: ~/Documents/kubsu-folder/cross-platform-programming-pra...
xom9chok37@ASUS:~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$
javac JavaApplication1.java
xom9chok37@ASUS:~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$
java JavaApplication1
Hello World!
2+2=4
xom9chok37@ASUS:~/Documents/kubsu-folder/cross-platform-programming-practice/lab1-cli$
```

Рисунок 4 — Выполним программу *Калькулятор*



```
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practic...
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1-cli$
touch SimpleFrame.java
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1-cli$
```

Рисунок 5 — Создадим файл *SimpleFrame.java*



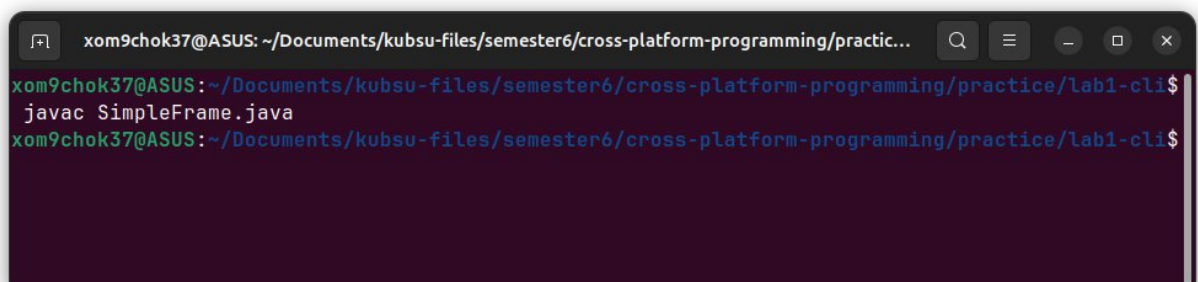
```
Open ▾ [icon] SimpleFrame.java
~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1-cli

import java.awt.*;
import java.awt.event.*;

// Создаем наш класс, наследуясь от стандартного класса Frame
class SimpleFrame extends Frame {
    SimpleFrame(String s) {
        super(s);
        // Передаем название окна в родительский класс
        setSize(400, 150); // Устанавливаем размер окна
        setVisible(true); // Окно становится видимым
        // Добавляем слушателя, который будет реагировать на кнопку
        // закрытия окна
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent ev) {
                dispose(); // Закрываем окно
                System.exit(0); // Выходим из программы
            }
        });
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent ev) {
                dispose();
                System.exit(0);
            }
        });
    }
}

// Создаем экземпляр нашего класса в главном классе
public static void main(String[] args) {
    new SimpleFrame(" Моя программа");
}
}
```

Рисунок 6 — Напишем код программы



```
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practic...
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1-cli$
javac SimpleFrame.java
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1-cli$
```

Рисунок 7 — Скомпилируем пример 1.1

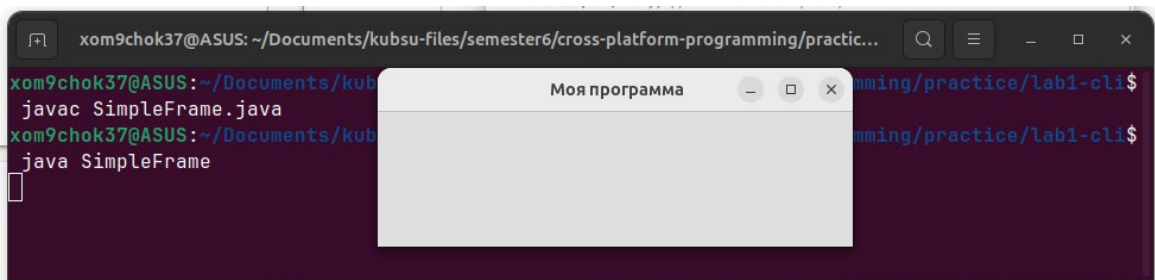


Рисунок 8 — Запустим пример 1.1

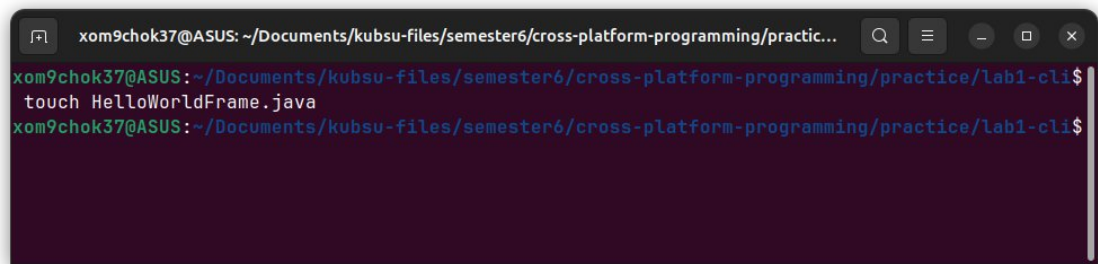


Рисунок 9 — Создадим файл *HelloWorldFrame.java*

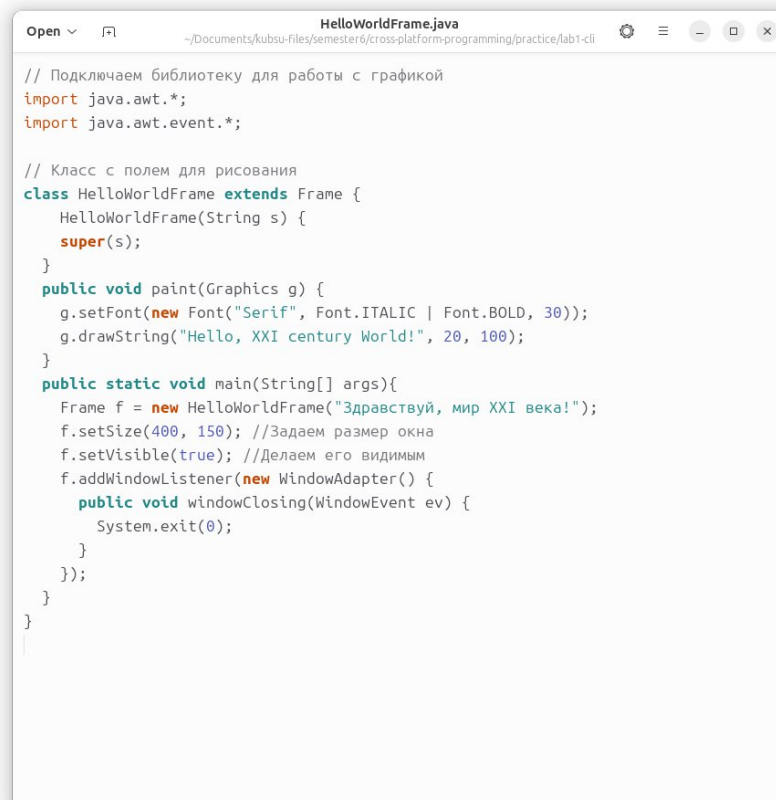


Рисунок 10 — Напишем код программы из примера 1.2

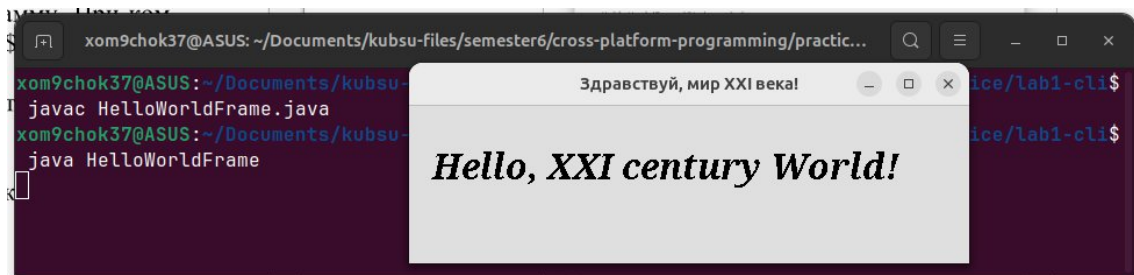


Рисунок 11 — Скомпилируем и запустим программу

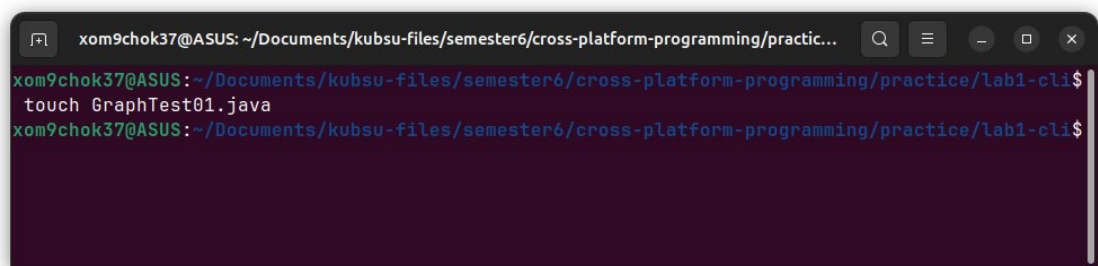


Рисунок 12 — Создадим файл *GraphTest01.java*

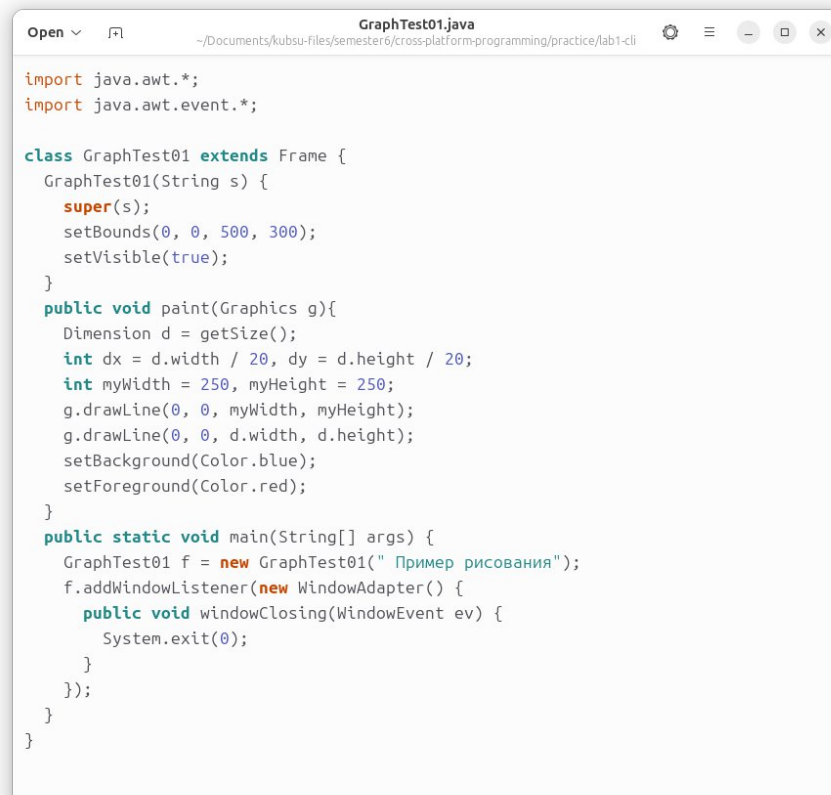


Рисунок 13 — Напишем код программы из примера 1.3



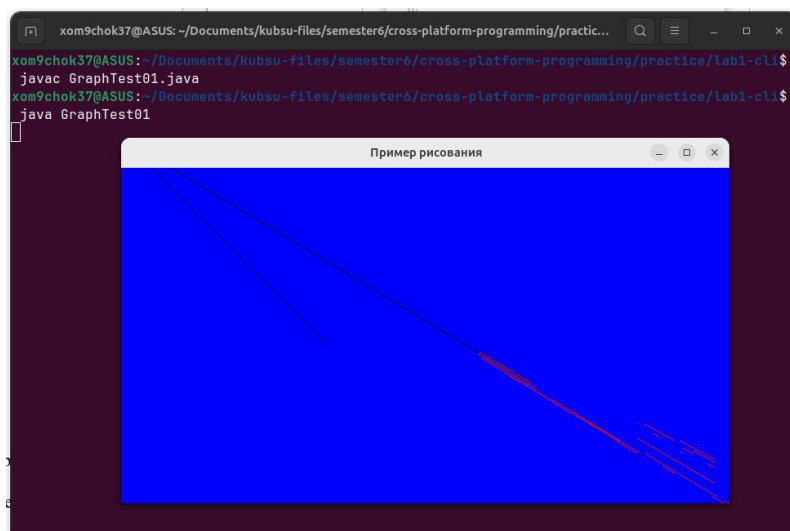


Рисунок 14 — Скомпилируем и запустим программу

2. Нарисуем круг, квадрат, картинку из примитивных фигур и линий (рисунок 15).

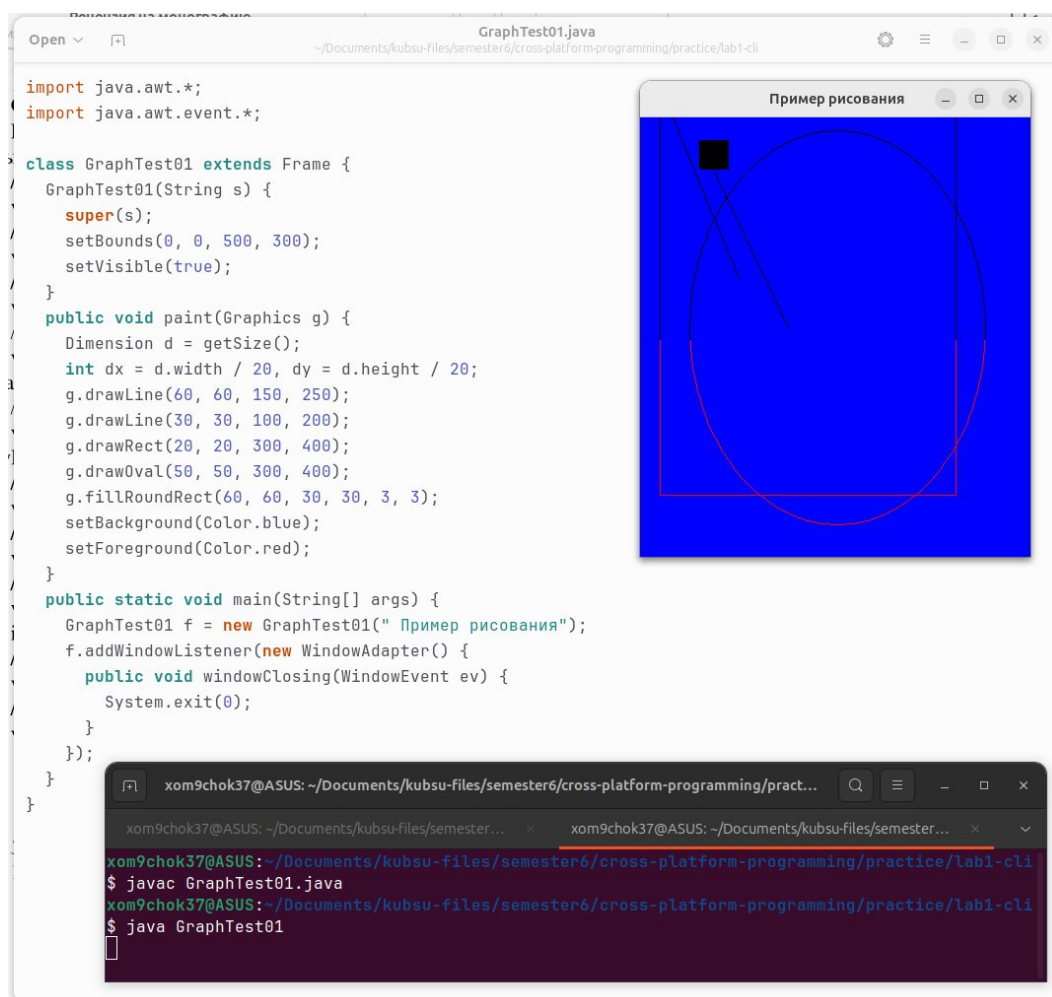
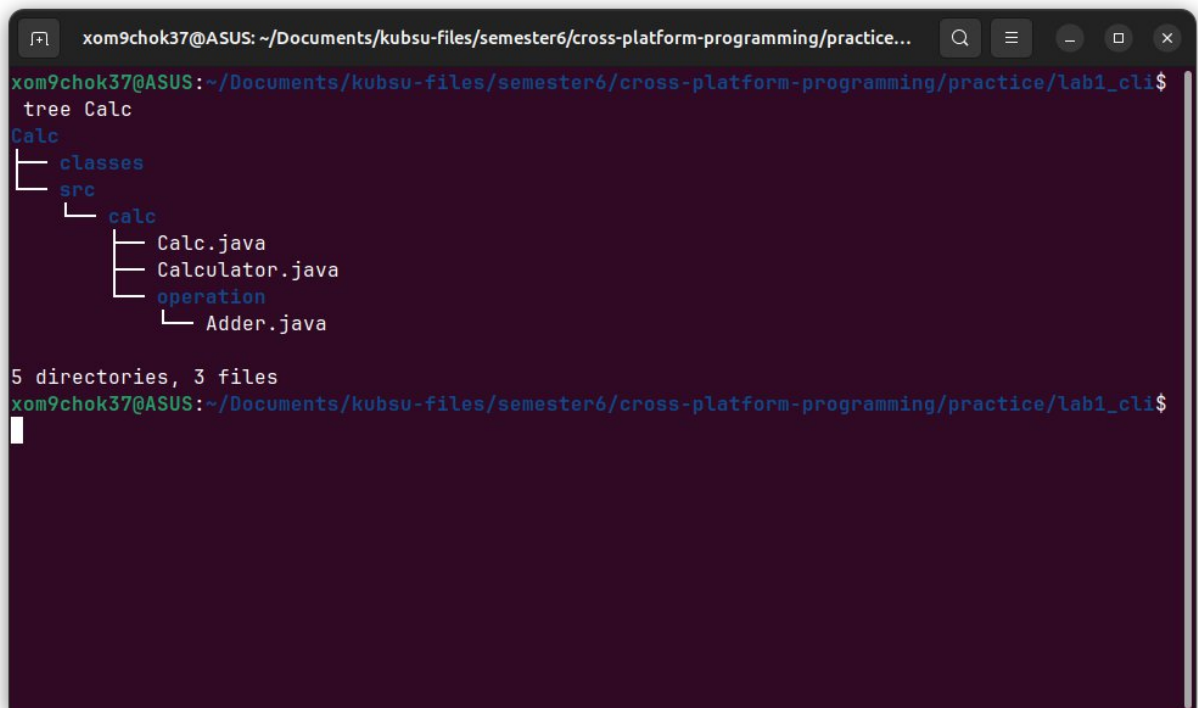


Рисунок 15 — Скомпилируем и запустим программу, рисующую фигуры

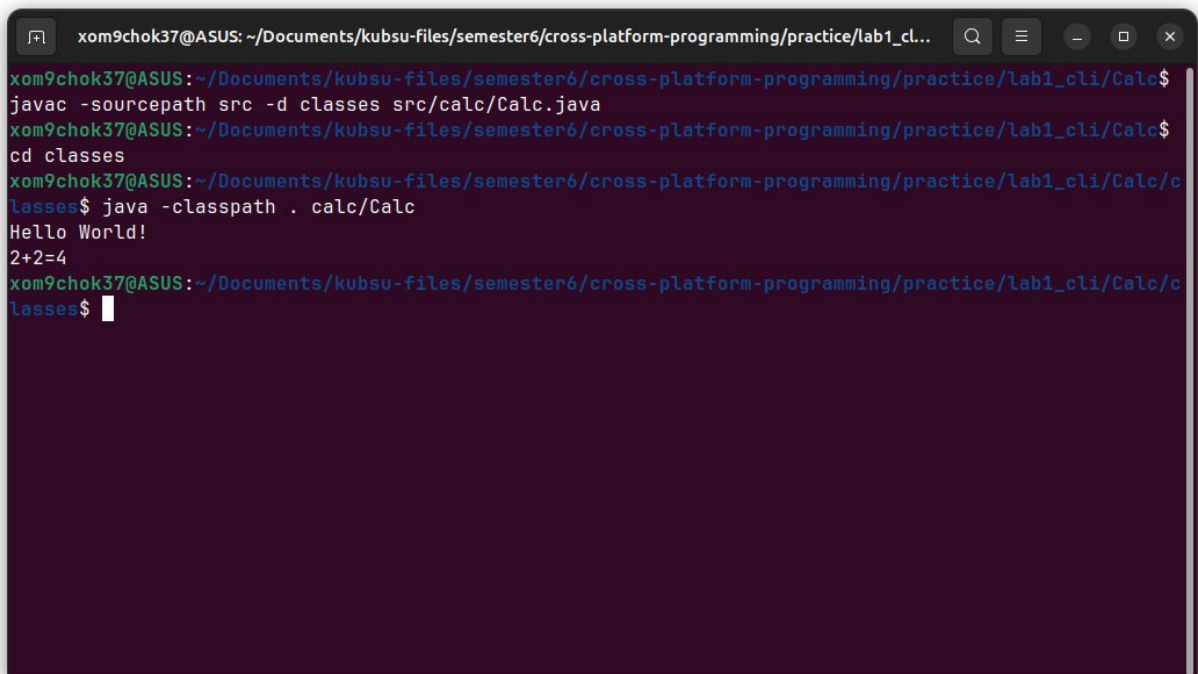
### 3. Создаём структуру приложения и компилируем (рисунки 16-17).



```
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice...
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli$
tree Calc
Calc
├── classes
├── src
│   └── calc
│       ├── Calc.java
│       ├── Calculator.java
│       └── operation
│           └── Adder.java

5 directories, 3 files
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli$
```

Рисунок 16 — Строим структуру



```
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli...
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli/Calc$
javac -sourcepath src -d classes src/calc/Calc.java
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli/Calc$
cd classes
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli/Calc/c
lasses$ java -classpath . calc/Calc
Hello World!
2+2=4
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli/Calc/c
lasses$
```

Рисунок 17 — Скомпилируем и запустим программу



4. Добавляем дополнительные классы, скомпилируем и запустим программу (рисунки 18-20).

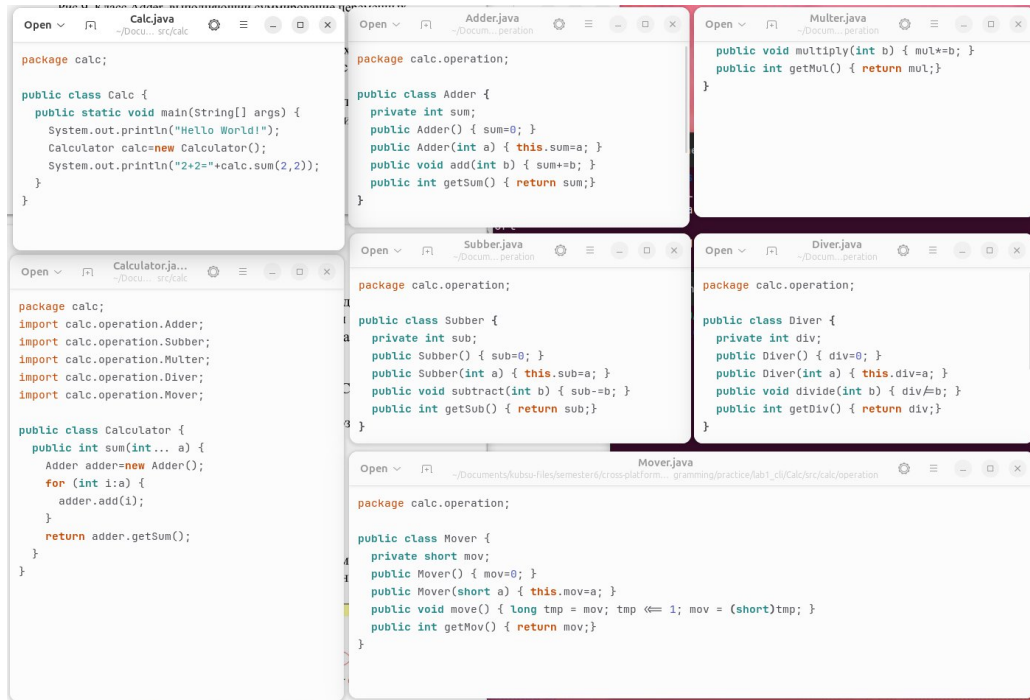


Рисунок 18 — Создаём новые классы

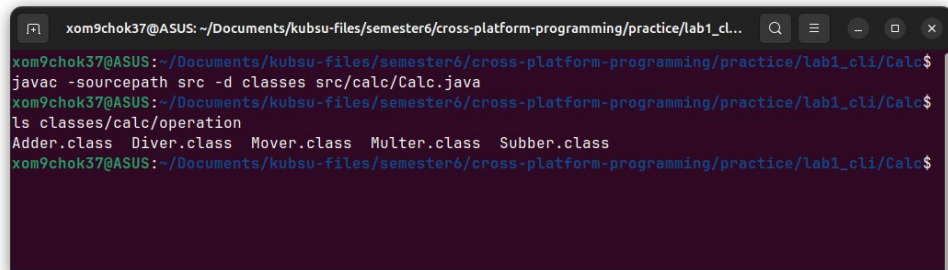


Рисунок 19 — Скомпилируем программу

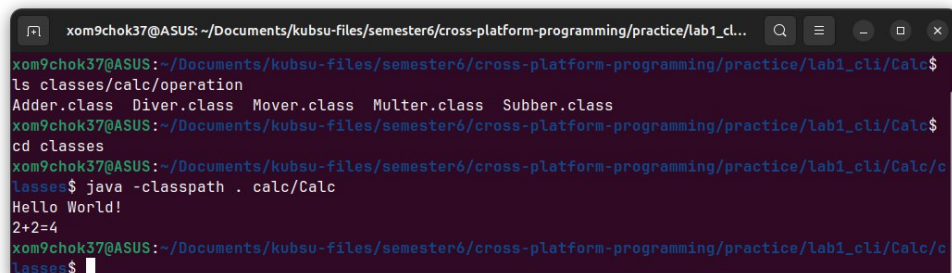
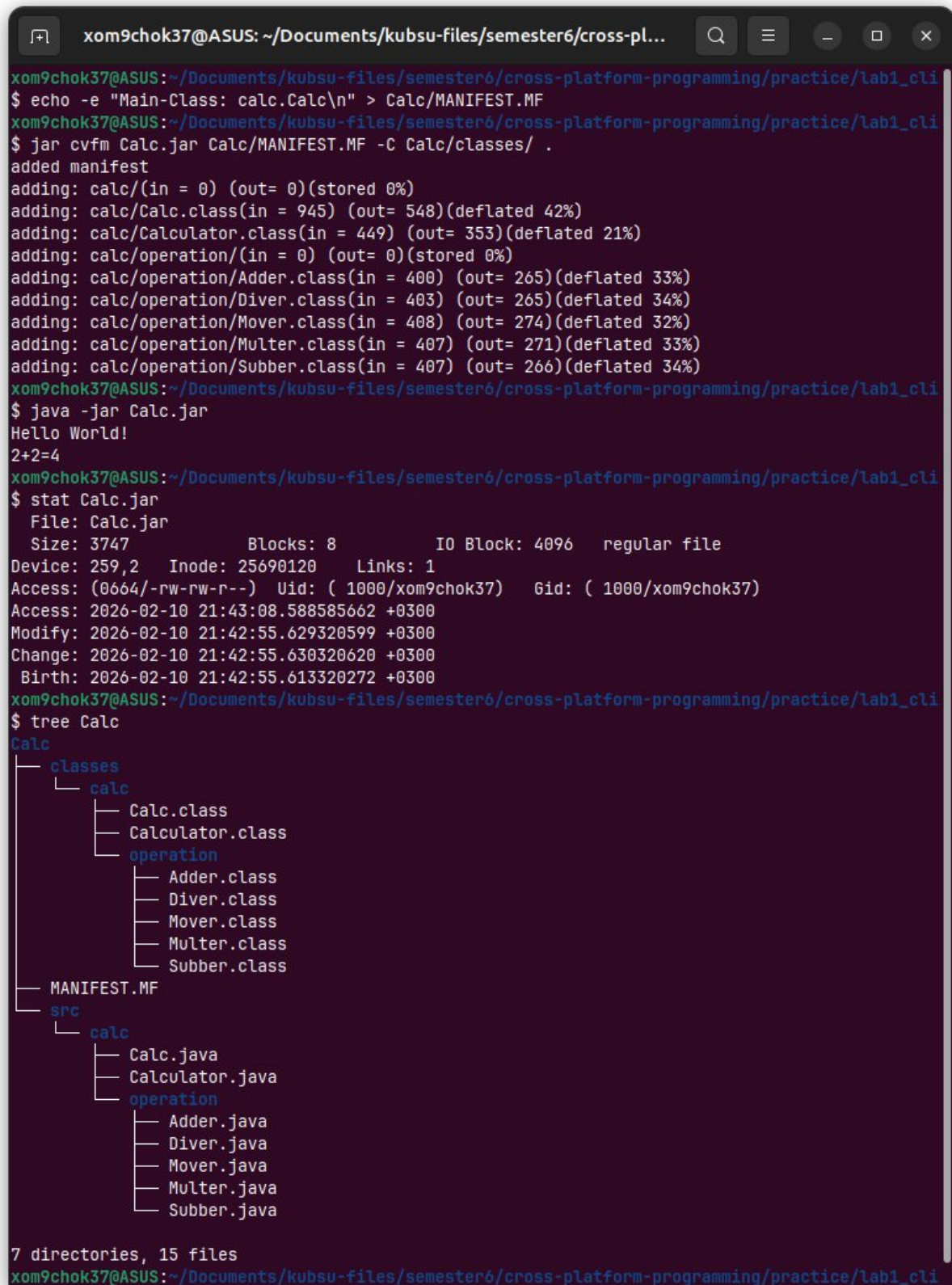


Рисунок 20 — Запустим программу

5. Добавим манифест в корневой каталог нашей программы, затем соберём программу в *JAR* и запустим её (рисунок 21).



```
xom9chok37@ASUS: ~/Documents/kubsu-files/semester6/cross-pl...
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli
$ echo -e "Main-Class: calc.Calc\n" > Calc/MANIFEST.MF
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli
$ jar cvfm Calc.jar Calc/MANIFEST.MF -C Calc/classes/ .
added manifest
adding: calc/(in = 0) (out= 0)(stored 0%)
adding: calc/Calc.class(in = 945) (out= 548)(deflated 42%)
adding: calc/Calculator.class(in = 449) (out= 353)(deflated 21%)
adding: calc/operation/(in = 0) (out= 0)(stored 0%)
adding: calc/operation/Adder.class(in = 400) (out= 265)(deflated 33%)
adding: calc/operation/Diver.class(in = 403) (out= 265)(deflated 34%)
adding: calc/operation/Mover.class(in = 408) (out= 274)(deflated 32%)
adding: calc/operation/Multer.class(in = 407) (out= 271)(deflated 33%)
adding: calc/operation/Subber.class(in = 407) (out= 266)(deflated 34%)
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli
$ java -jar Calc.jar
Hello World!
2+2=4
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli
$ stat Calc.jar
  File: Calc.jar
  Size: 3747          Blocks: 8          IO Block: 4096   regular file
Device: 259,2   Inode: 25690120   Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/xom9chok37)   Gid: ( 1000/xom9chok37)
Access: 2026-02-10 21:43:08.588585662 +0300
Modify: 2026-02-10 21:42:55.629320599 +0300
Change: 2026-02-10 21:42:55.630320620 +0300
 Birth: 2026-02-10 21:42:55.613320272 +0300
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli
$ tree Calc
Calc
├── classes
│   └── calc
│       ├── Calc.class
│       ├── Calculator.class
│       └── operation
│           ├── Adder.class
│           ├── Diver.class
│           ├── Mover.class
│           ├── Multer.class
│           └── Subber.class
└── MANIFEST.MF
src
└── calc
    ├── Calc.java
    ├── Calculator.java
    └── operation
        ├── Adder.java
        ├── Diver.java
        ├── Mover.java
        ├── Multer.java
        └── Subber.java

7 directories, 15 files
xom9chok37@ASUS:~/Documents/kubsu-files/semester6/cross-platform-programming/practice/lab1_cli
```

Рисунок 21 — Создаём манифест и собираем программу