**Постановка задачи.**

Разработать программное средство, в котором присутствовали бы некоторые критерии и примитивы качественного программного обеспечения. Сделать вывод о проделанной работе.

Имеется арифметический квадрат:

1 1 1 1 1 1 …

1 2 3 4 5 6 …

1 3 6 10 15 21 …

1 4 10 20 35 56 …

Найдите n(i,j) член квадрата.

**Ход работы**

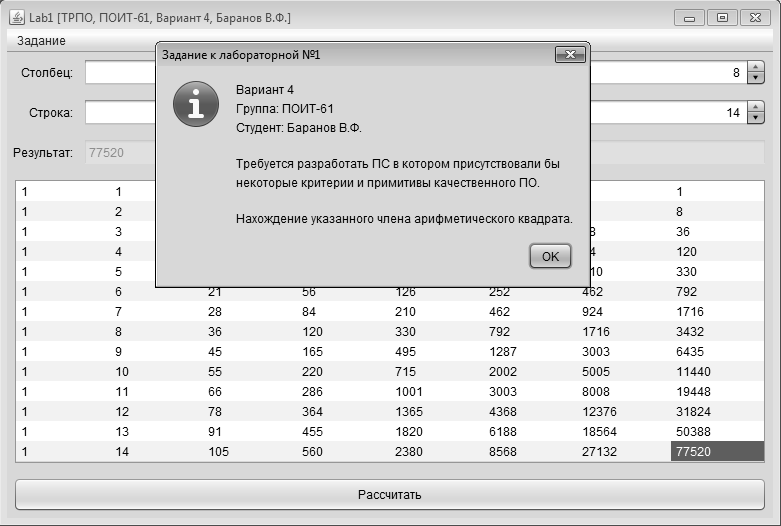
***Описание приложения.***

Для решения поставленной задачи, было разработано desktop-приложение на платформе Java. Для создания графического интерфейса пользователя (GUI) использовался функционал библиотеки Swing и возможности GUI-редактора среды разработки NetBeans.

Приложение состоит из 4 классов: Runner.java, Constants.java, MainWindow.java, SquareCalculateHelper.java.

* Класс MainWindow.java является наследником класса JFrame.java и главным окном приложения (Рисунок 1). В нём реализованы методы обработки действий пользователя и отображение результатов работы приложения.
* SquareCalculateHelper.java предоставляет метод вычисления арифметического квадрата.
* Из класса Runner.java осуществляет запуск приложения, создание и инициализация классов главного окна приложения и класса helper'а для вычислений.
* В классе Constants.java собраны константы и строковые литералы приложения.

Для запуска приложения требуется установленная виртуальная машина Java Runtime Environment (JRE) версии 8 или выше. Она распространяется свободно и для большинства платформ может быть загружена с сайта Oracle.

******

***Использованные критерии и примитивы качественного ПО.***

При разработке программного средства использовался документ Code Conventions for the Java Programming Language, расположенный на сайте Oracle. В нём описаны основные соглашения по программированию на языке Java, которым следуют разработчики Sun и рекомендуют к использованию всем Java-разработчикам. Они покрывают правила именования файлов, классов, атрибутов и методов, форматирование и стиль написания кода, а также правила комментирования и документирования кода и основные практики программирования, представленные соответствующими примерами.

В связи с большим объёмом, ниже будут перечислены лишь основные пункты соглашения, использованные при разработке:

Написание названий классов приложения осуществляется в CamelCase нотации с первой буквой первого слова в верхнем регистре; констант и строковых литералов в UpperCase с нижним подчёркиванием в качестве разделителя; названий методов, полей и переменных в CamelCase с первым словом, начинающимся в нижнем регистре.

Для комментирования классов, полей, методов и параметров методов применяется функционал генератора документации javaDoc.

Названия классов, полей, методов, констант и переменных обладают смысловой нагрузкой и позволяют понять своё назначение из названия.

В исходном коде приложения использовано минимальное количество неименованных констант, во избежание анти-паттерна проектирования "magic number".

Также, за исключением сгенерированной редактором IDE части класса MainWindow, используется порядок размещения методов и полей класса в зависимости от области видимости: сначала большая видимость, затем меньшая.

Большинство строк кода приложения, за исключением сгенерированной редактором IDE части класса MainWindow, написаны с учётом рекомендации на длину строк в 80 символов.

**Вывод**

Разработанное программное средство полностью выполняет поставленную задачу. При разработке использовалось соглашение по программированию на языке Java, принятое разработчиками языка и которое рекомендуют к использованию всем Java-разработчикам. Они покрывают правила именования файлов, классов, атрибутов и методов, форматирование и стиль написания кода, а также правила комментирования и документирования кода и основные практики программирования.