ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИНСТИТУТ КОМПЬЮТЕРНЫХ СИСТЕМ

Кафедра «Информационные системы и технологии»

**Лабораторная работа №1**

по дисциплине: «Объектно-ориентированное программирование»

на тему: «Разработка консольных приложений»

Вариант 13

**Выполнил:**

ст. гр. НАД-191

Краковский В.А.

**Проверили:**

д. Рудниченко М.Д.

ст. пр. Павлов О.А.

Одесса 2020

Оглавление

[ВВЕДЕНИЕ 3](#__RefHeading___Toc131_1832763852)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#__RefHeading___Toc133_1832763852)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 5](#__RefHeading___Toc135_1832763852)

[ВЫВОД 8](#__RefHeading___Toc137_1832763852)

# ВВЕДЕНИЕ

Целью данной лабораторной работы является изучение процесса разработки консольных приложений на языке программирования Java с помощью IDE для реализации простейшего приложения.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Консольные приложения JAVA представляют собой созданный и откомпилированный программистом класс, содержащий точку входа.

**Класс System**

Класс System содержит набор полезных статических методов и полей системного уровня. Экземпляр этого класса не может быть создан или получен.

Наиболее широко используемой возможностью, предоставляемой System, является стандартный вывод, доступный через переменную System.out.

**Класс String**

Класс String содержит основные методы для работы со строками:

concat(String s) или + – слияние строк;

equals(Object ob), equalsIgnoreCase(String s) – сравнение строк с учетом и без учета регистра;

compareTo(String s), compareToIgnoreCase (String s) – лексикографическое сравнение строк с учетом и без учета регистра;

contentEquals(StringBuffer ob) – сравнение строки и содержимого объекта типа StringBuffer;

charAt(int n)– извлечение из строки символа с указанным номером (нумерация с нуля);

substring(int n, int m)- извлечение из строки подстроки длины m-n, начиная с позиции n;

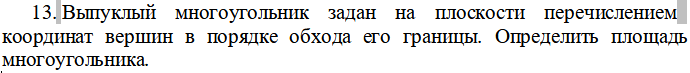
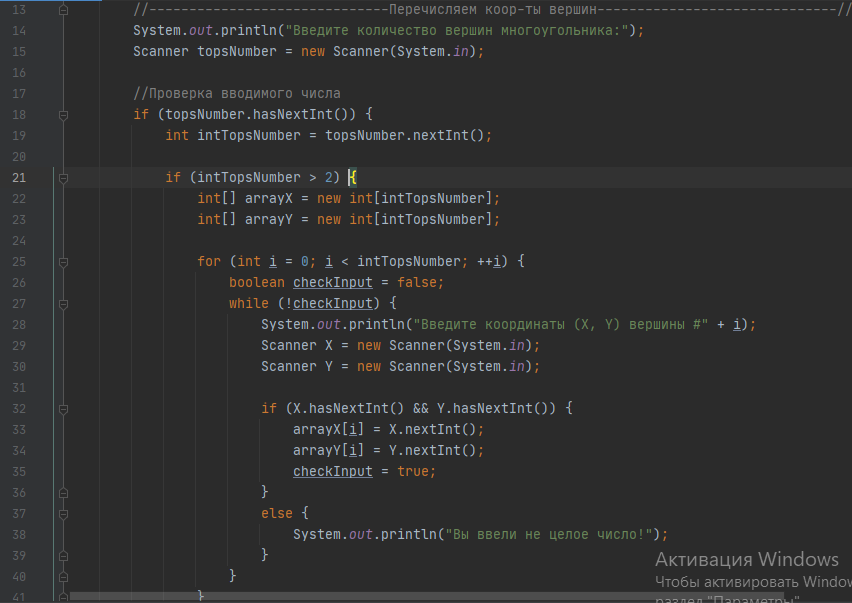
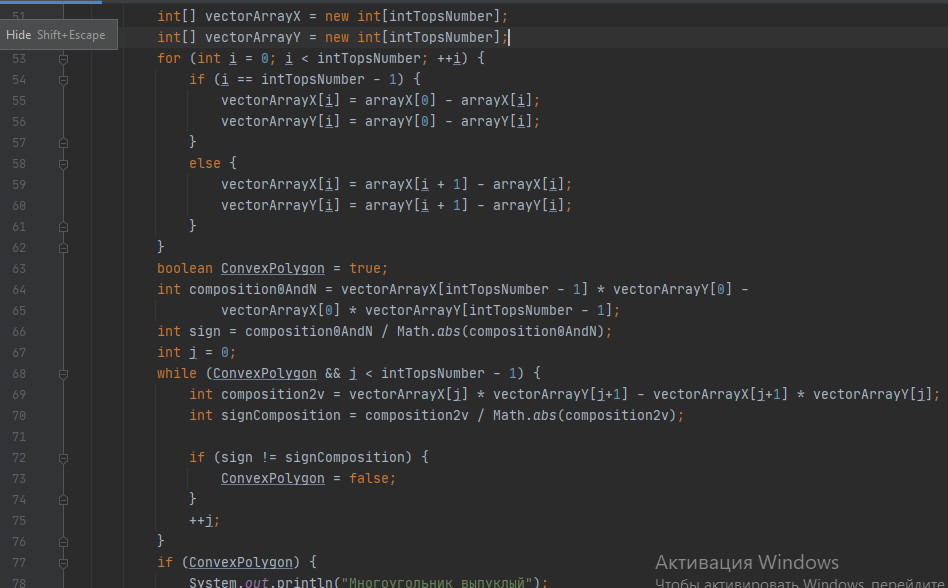
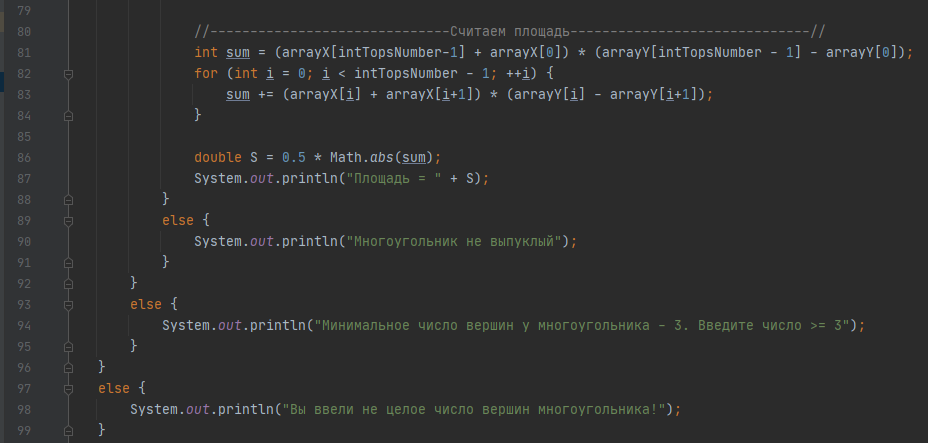
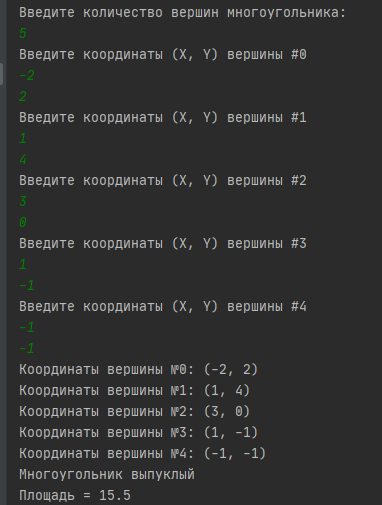
length() – определение длины строки;

valueOf(объект) – преобразование примитивного объекта к строке;

toUpperCase() / toLowerCase() – преобразование всех символов вызывающей строки в верхний/нижний регистр;

replace(char с1, char с2) – замена в строке всех вхождений первого символа вторым символом;

# ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Условие
2. Вводим координаты с клавиатуры:
3. Проверяем многоугольник на выпуклость:
4. Считаем искомую площадь:
5. Результат:

# ВЫВОД

На этой лабораторной работе я узнал каким образом создавать простейшее приложение с помощью консоли и простейших операций на языке программирования Java. Как результат, была реализована программа для подсчёта площади выпуклого многоугольника, которую возможно использовать в соответствующих математических подсчётах.

# ЛИТЕРАТУРА

1. Выпуклый многоугольник [Электронный ресурс] - https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D0%BF%D1%83%D0%BA%D0%BB%D1%8B%D0%B9\_%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA
2. Выпуклость многоугольника [Электронный ресурс] - <https://tux.org.ua/vy-puklost-mnogougol-nika/>