Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.01

Выполнила: Горелова Э.А

Группа: ПР-31

Преподаватель: Мирошниченко Г.В.

2024

# Содержание

[Содержание 2](#_Toc183354761)

[1. Задание 4. Библиотека классов. Модульные тесты 3](#_Toc183354762)

[1.1. Описание задачи: 3](#_Toc183354763)

[1.2. Описание разработанных функций 3](#_Toc183354764)

[1.3. Алгоритм решения 5](#_Toc183354765)

[1.4. Используемые библиотеки 7](#_Toc183354766)

[1.5. Тестовые случаи 8](#_Toc183354767)

[1.6. Используемые инструменты 21](#_Toc183354768)

[Язык написания данной программы- C# 21](#_Toc183354769)

[Среда разработки – WindowsForms C# 21](#_Toc183354770)

[1.7. Описание пользовательского интерфейса 21](#_Toc183354771)

[1.8. Результат 22](#_Toc183354772)

# 1. Задание 4. Библиотека классов. Модульные тесты

## 1.1. Описание задачи:

В консоли решить задание и написать по 20 модульных тестов по заданию:

**Вариант 1:** «Стоимость разговора». Петя Торопыжкин пользуется услугами оператора сотовой связи «НУТ» по тарифном плану «А ну-ка посчитай-ка!». Условия данного плана таковы: оплата соединения — a рублей, первая минута разговора бесплатна, минуты со второй по n-ю включительно стоят по b рублей, после n-й минуты разговор снова бесплатен. Очередное общение с девушкой заняло у Пети k минут. На сколько рублей стало меньше на счёте у Пети?

**Вариант 3:** «Большая клумба». Класс Пети Торопыжкина решил разбить перед школой две клумбы, каждая в виде прямоугольника со сторонами, параллельными осям Школьной Декартовой Системы Координат. Мальчики разобьют клумбу с двумя противоположными вершинами в точках (x1m,y1m) и (x2m,y2m), а девочки — с координатами (x1d,y1d) и (x2d,y2d). Но так как выбор делался независимо, то клумбы могут пересекаться. Найдите площадь, которую надо будет вскопать классу Пети Торопыжкина под клумбы. Формат ввода: В первой строке файла через пробел заданы четыре целых числа x1m, y1m, x2m, y2m — координаты вершин клумбы мальчиков. Во второй строке так же через пробел заданы четыре целых числа x1d, y1d, x2d, y2d — координаты вершин клумбы девочек. Все координаты — целые числа, по модулю не превосходящие 104.

## Описание разработанных функций

**Вариант 1:** Функция Calculate предназначена для вычисления целочисленного значения на основе входных данных *(Рис.1).*

**Входные данные:**

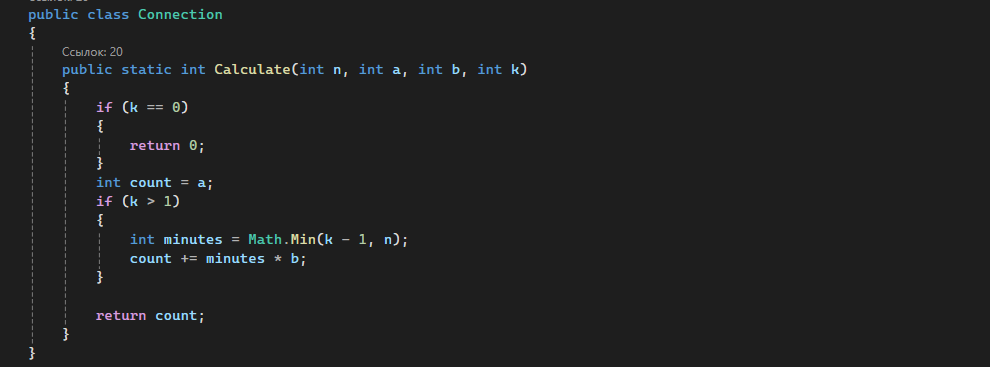
a — стоимость соединения (в рублях).

b — стоимость минуты разговора со второй по n-ю (в рублях).

n — количество минут, по истечении которых разговор снова становится бесплатным (включительно).

k — общее время разговора в минутах (включая бесплатные минуты).

**Выходные данные:** метод возвращает целое число-int.



*Рис.1.*Функция Calculate

**Вариант 3:** Функция CalculateFlowerBeds предназначена для вычисления площади клумб на основе их координат *(Рис.2).*

**Входные данные:**

**Первая клумба**:

x1m: координата x нижнего левого угла первой клумбы.

y1m: координата y нижнего левого угла первой клумбы.

x2m: координата x верхнего правого угла первой клумбы.

y2m: координата y правого верхнего угла первой клумбы.

**Вторая клумба**:

x1d: координата x нижнего левого угла второй клумбы.

y1d: координата y нижнего левого угла второй клумбы.

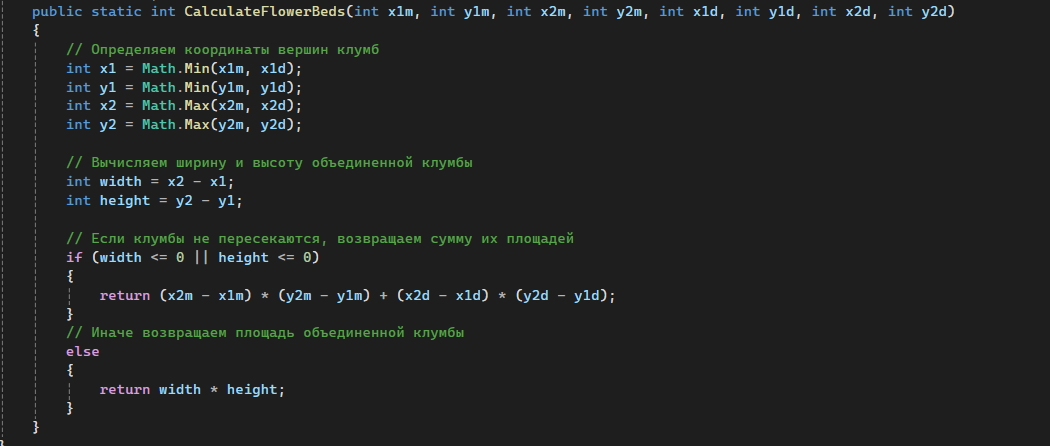
x2d: координата x верхнего правого угла второй клумбы.

y2d: координата y правого верхнего угла второй клумбы.

**Выходные данные:**

Функция возвращает целое число, представляющее площадь объединенных цветочных клумб:

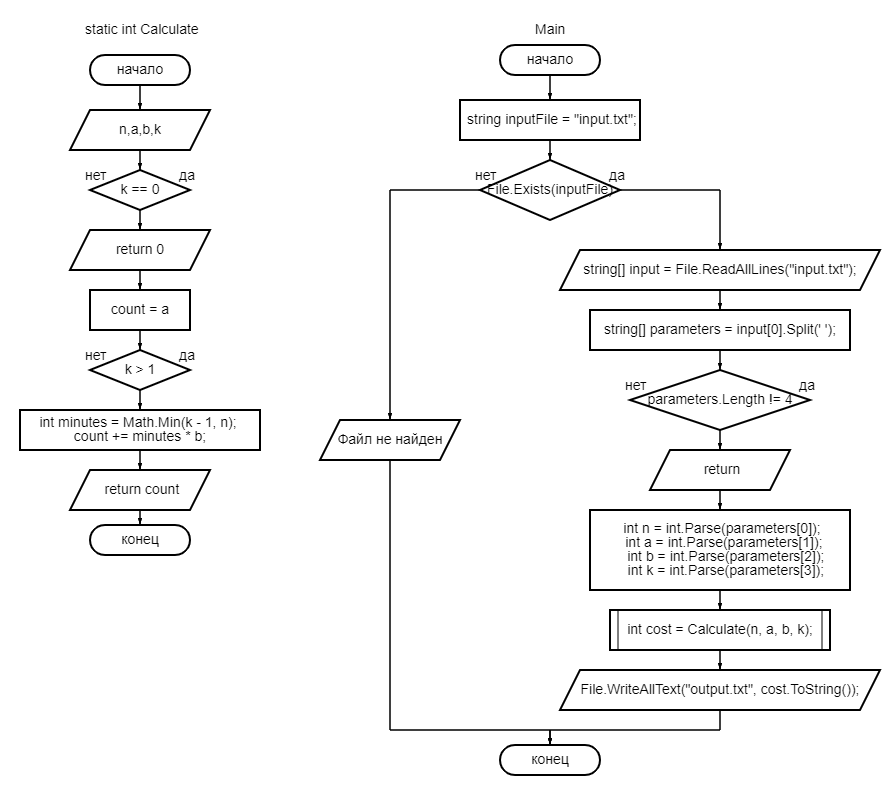
* Если две клумбы не пересекаются, возвращается сумма их отдельных площадей.
* Если они перекрываются, возвращается площадь прямоугольника, который охватывает обе клумбы.



*Рис.2.*Функция CalculateFlowerBeds

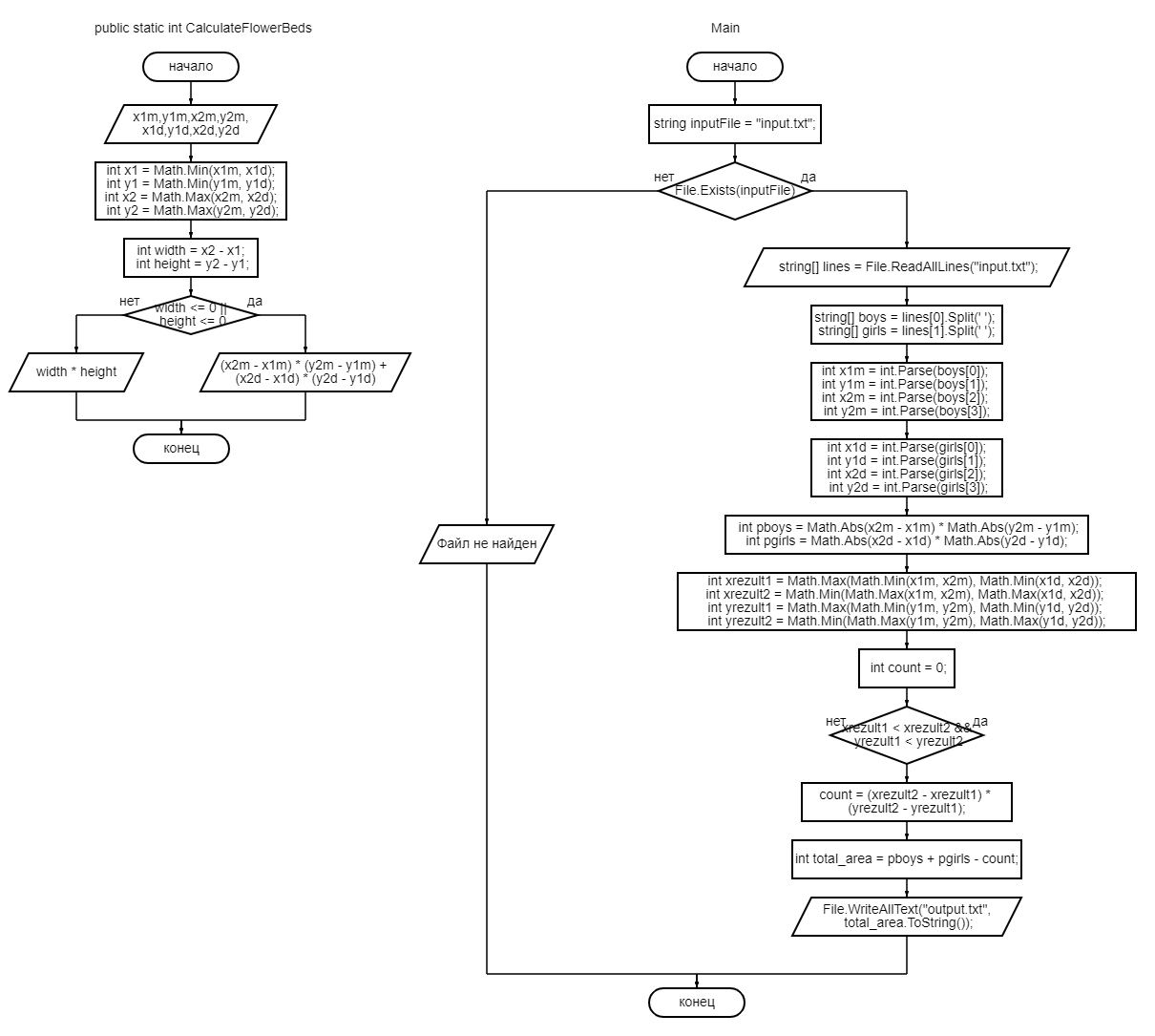
## 1.3. Алгоритм решения

**Вариант 1:**



*Рис.3.* Блок-схема

**Вариант 3:**



*Рис.4.* Блок-схема

## 1.4. Используемые библиотеки

**1 вариант:**

System.IO-для работы с файлами.

**3 вариант:**

System.IO-для работы с файлами.

## 1.5. Тестовые случаи

**1 вариант:**

1 тест: TestCalculateZeroMinute *(Рис.5)*

Тест для проверки нулевого времени, результат будет равен 0, так как нет затрат на соединение.

**Входные данные:**

n=5, a=2, b=3, k=0

**Ожидаемый результат:** 0

*Рис.5.* TestCalculateZeroMinute

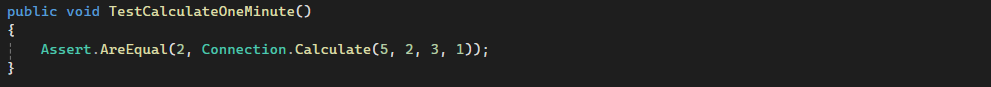
2 тест: TestCalculateOneMinute *(Рис.6)*

Тест для проверки расчёта за 1 минуту, результат равен начальному значению a, так как нет дополнительных минут для начисления.

**Входные данные:**

n=5, a=2, b=3, k=1

**Ожидаемый результат:** 2



*Рис.6.* TestCalculateOneMinute

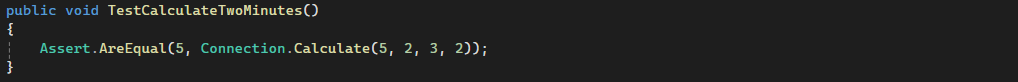
3 тест: TestCalculateTwoMinutes *(Рис.7)*

Тест для проверки расчёта за 2 минуты, результат равен a + b, где b — стоимость за минуту.

**Входные данные:**

n=5, a=2, b=3, k=3

**Ожидаемый результат:** 5

*Рис.7.* TestCalculateTwoMinutes

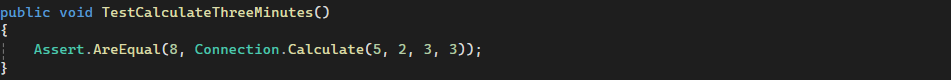
4 тест: TestCalculateThreeMinutes *(Рис.8)*

Тест для проверки расчёта за 3 минуты, результат равен a + 2 \* b, что учитывает две дополнительные минуты.

**Входные данные:**

n=5, a=2, b=3, k=3

**Ожидаемый результат:** 8

****

*Рис.8.* TestCalculateThreeMinutes

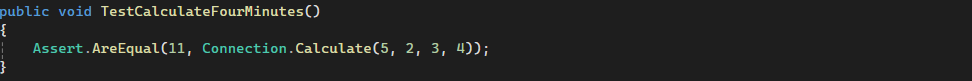
5 тест: TestCalculateFourMinutes *(Рис.9)*

Тест для проверки расчёта за 4 минуты, результат равен a + 3 \* b.

**Входные данные:**

n=5, a=2, b=3, k=4

**Ожидаемый результат:** 11

*Рис.9.* TestCalculateFourMinutes

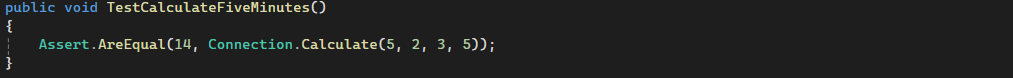
6 тест: TestCalculateFiveMinutes *(Рис.10)*

Тест для проверки расчёта за 5 минут, результат равен a + 4 \* b.

**Входные данные:**

n=5, a=2, b=3, k=5

**Ожидаемый результат:** 14

*Рис.10.* TestCalculateFiveMinutes

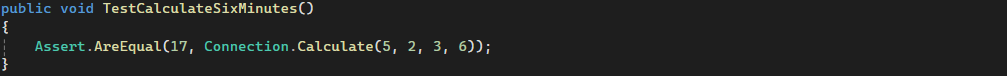
7 тест: TestCalculateSixMinutes *(Рис.11)*

Тест для проверки расчёта за 6 минут, результат равен a + 5 \* b, что соответствует максимальному количеству минут.

**Входные данные:**

n=5, a=2, b=3, k=6

**Ожидаемый результат:** 17

*Рис.11.* TestCalculateSixMinutes

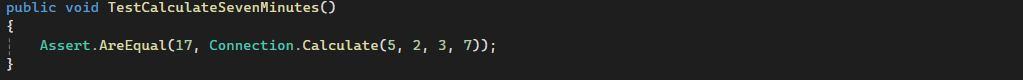
8 тест: TestCalculateSevenMinutes *(Рис.12)*

Тест для проверки расчёта за 7 минут, результат остаётся равным a + 5 \* b, так как n ограничивает количество минут.

**Входные данные:**

n=5, a=2, b=3, k=7

**Ожидаемый результат:** 17

*Рис.12.* TestCalculateSevenMinutes

9 тест: TestCalculateMaxMinutes *(Рис.13)*

Тест проверяет, что при большом значении k (например, 100) результат остаётся равным a + 5 \* b, так как n ограничивает количество минут.

**Входные данные:**

n=5, a=2, b=3, k=100

**Ожидаемый результат:** 17

*Рис.13.* TestCalculateMaxMinutes

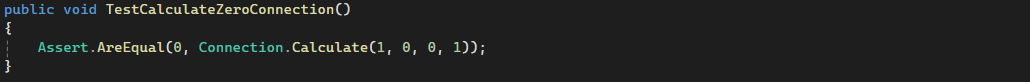
10 тест: TestCalculateZeroConnection *(Рис.14)*

Тест проверяет, что если начальная стоимость соединения равна 0, то результат также будет равен 0, даже если есть минуты.

**Входные данные:**

n=1, a=0, b=0, k=1

**Ожидаемый результат:** 0

*Рис.14.* TestCalculateZeroConnection

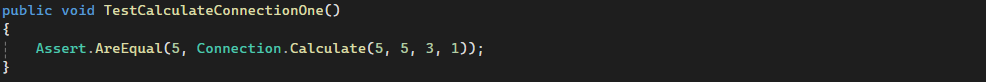
11 тест: TestCalculateConnectionOne *(Рис.15)*

Тест проверяет, что если разговор длится 1 минуту, а стоимость соединения a больше 0, то итоговая стоимость равна a.

**Входные данные:**

n=5, a=5, b=3, k=1

**Ожидаемый результат:** 5

*Рис.15.* TestCalculateConnectionOne

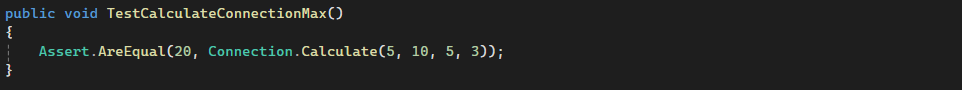
12 тест: TestCalculateConnectionMax *(Рис.16)*

Тест проверяет, что при 3 минутах и высокой стоимости соединения результат соответствует ожидаемому значению.

**Входные данные:**

n=5, a=10, b=3, k=3

**Ожидаемый результат:** 20

*Рис.16.* TestCalculateConnectionMax

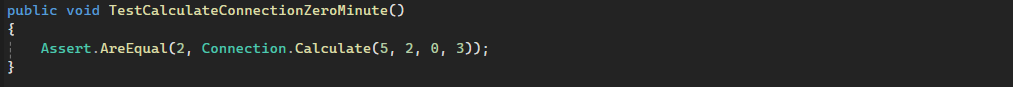
13 тест: TestCalculateConnectionZeroMinute *(Рис.17)*

Тест проверяет, что если стоимость в минуту равна 0, то результат будет равен начальному значению a.

**Входные данные:**

n=5, a=2, b=0, k=3

**Ожидаемый результат:** 2

*Рис.17.* TestCalculateConnectionZeroMinute

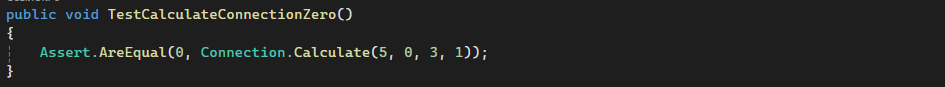
14 тест: TestCalculateConnectionZero *(Рис.18)*

Тест проверяет, что если стоимость соединения равна 0, то итоговая стоимость 1 минуты разговора также равна 0.

**Входные данные:**

n=5, a=2, b=0, k=3

**Ожидаемый результат:** 0

*Рис.18.* TestCalculateConnectionZero

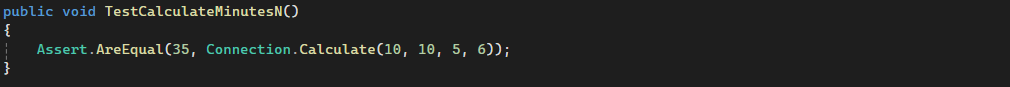
15 тест: TestCalculateMinutesN *(Рис.19)*

Тест проверяет, что если минут больше, чем n, то стоимость остаётся равной a + (n-1) \* b, так как после n-й минуты разговор бесплатен.

**Входные данные:**

n=10, a=10, b=5, k=6

**Ожидаемый результат:** 35

*Рис.19.* TestCalculateMinutesN

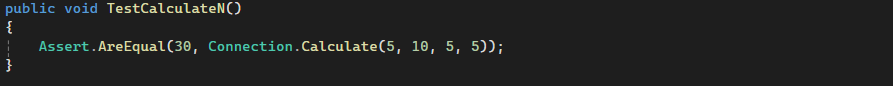
16 тест: TestCalculateN *(Рис.20)*

Тест проверяет, что если минут ровно n, то стоимость равна a + (n-1) \* b.

**Входные данные:**

n=5, a=10, b=5, k=5

**Ожидаемый результат:** 30

*Рис.20.* TestCalculateN

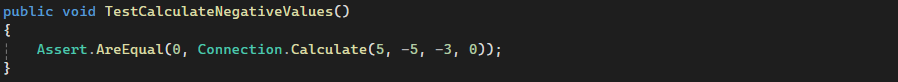
17 тест: TestCalculateNegativeValues *(Рис.21)*

Тест проверяет, что если присутствуют отрицательные значения, итоговая стоимость равна 0.

**Входные данные:**

n=5, a=-5, b=-3, k=0

**Ожидаемый результат:** 0

*Рис.21.* TestCalculateNegativeValues

18 тест: TestCalculateNegativeMinute *(Рис.22)*

Тест проверяет, что если стоимость минуты отрицательная, то итоговая стоимость не меняется и равна стоимости соединения a.

**Входные данные:**

n=5, a=2, b=-3, k=1

**Ожидаемый результат:** 2

*Рис.22.* TestCalculateNegativeMinute

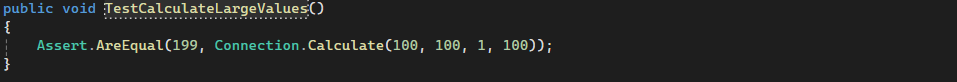
19 тест: TestCalculateLargeValues *(Рис.23)*

Тест проверяет, что при больших значениях входных параметров метод корректно рассчитывает итоговую стоимость.

**Входные данные:**

n=100, a=100, b=1, k=100

**Ожидаемый результат:** 199

*Рис.23.* TestCalculateLargeValues

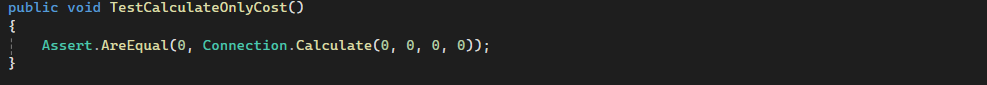
20 тест: TestCalculateOnlyCost *(Рис.24)*

Тест проверяет, что метод правильно обрабатывает крайние случаи, когда нет никаких значений для расчёта.

**Входные данные:**

n=0, a=0, b=0, k=0

**Ожидаемый результат:** 0

*Рис.24.* TestCalculateOnlyCost

**3 вариант:**

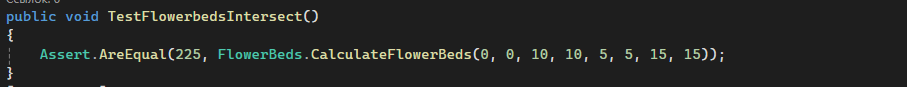
1 тест: TestFlowerbedsIntersect *(Рис.25)*

Тест проверяет, что клумбы пересекаются.

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=5, y1d=5, x2d=15, y2d=15

**Ожидаемый результат:** 225

*****Рис.25.* TestFlowerbedsIntersect

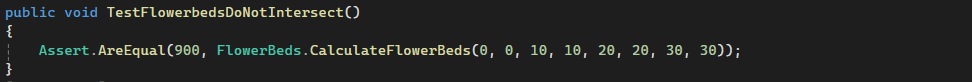
2 тест: TestFlowerbedsDoNotIntersect *(Рис.26)*

Тест проверяет, что клумбы не пересекаются

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=20, y1d=20, x2d=30, y2d=30

**Ожидаемый результат:** 900

*Рис.26.* TestFlowerbedsDoNotIntersect

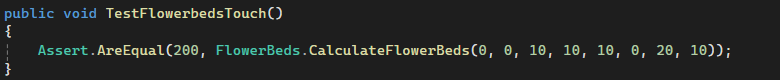
3 тест: TestFlowerbedsTouch *(Рис.27)*

Тест проверяет, что клумбы касаются друг друга

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=10, y1d=0, x2d=20, y2d=10

**Ожидаемый результат:** 200

*Рис.27.* TestFlowerbedsTouch

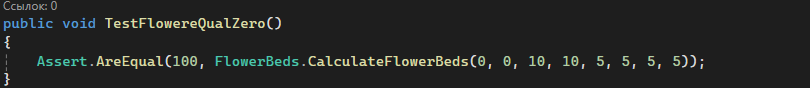
4 тест: TestFlowereQualZero *(Рис.28)*

Тест проверяет, что площадь одной из клумб равна нулю.

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=5, y1d=5, x2d=5, y2d=5

**Ожидаемый результат:** 100

*Рис.28.* TestFlowereQualZero

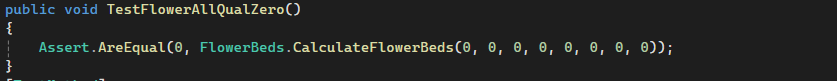
5 тест: TestFlowerAllQualZero *(Рис.29)*

Тест проверяет, что все координаты клумб равны нулю

**Входные данные:**

x1m=0, y1m=0, x2m=0, y2m=0, x1d=0, y1d=0, x2d=0, y2d=0

**Ожидаемый результат:** 0

*Рис.29.* TestFlowerAllQualZero

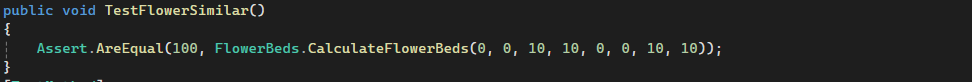
6 тест: TestFlowerSimilar *(Рис.30)*

Тест проверяет, что координаты клумб одинаковые

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=0, y1d=0, x2d=10, y2d=10

**Ожидаемый результат:** 100

*Рис.30.* TestFlowerSimilar

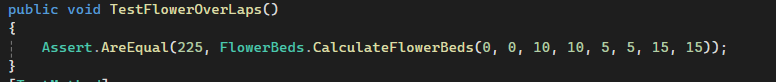
7 тест: TestFlowerOverLaps *(Рис.31)*

Тест проверяет, что полностью одна клумба перекрывает другую.

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=5, y1d=5, x2d=15, y2d=15

**Ожидаемый результат:** 225

*Рис.31.* TestFlowerOverLaps

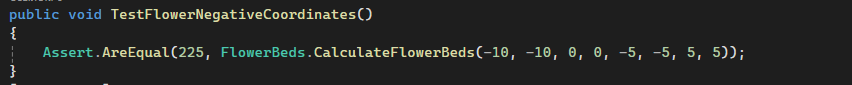
8 тест: TestFlowerNegativeCoordinates *(Рис.32)*

Тест проверяет, клумбы с отрицательными координатами.

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=5, y1d=5, x2d=15, y2d=15

**Ожидаемый результат:** 225

*Рис.32.* TestFlowerNegativeCoordinates

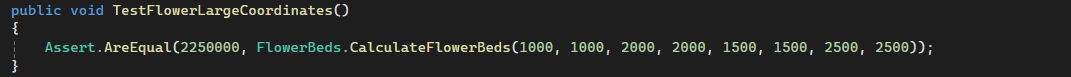
9 тест: TestFlowerLargeCoordinates *(Рис.33)*

Тест проверяет, клумбы с большими значениями

**Входные данные:**

x1m=1000, y1m=1000, x2m=2000, y2m=2000, x1d=1500, y1d=1500, x2d=2500, y2d=2500

**Ожидаемый результат:** 2250000

*Рис.33.* TestFlowerLargeCoordinates

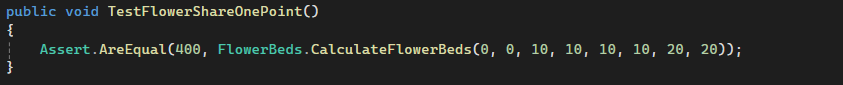
10 тест: TestFlowerShareOnePoints *(Рис.34)*

Тест проверяет, клумбы с одной общей точкой

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=10, y1d=10, x2d=20, y2d=20

**Ожидаемый результат:** 400

*Рис.34.* TestFlowerShareOnePoints

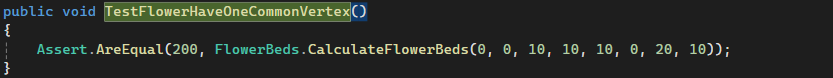
11 тест: TestFlowerHaveOneCommonVertex *(Рис.35)*

Тест проверяет, клумбы с одной общей вершиной

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=10, y1d=0, x2d=20, y2d=10

**Ожидаемый результат:** 200

*Рис.35.* TestFlowerHaveOneCommonVertex

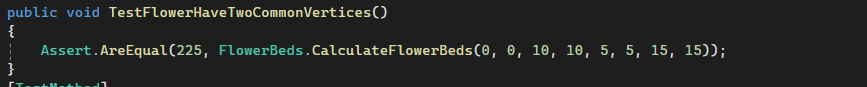
12 тест: TestFlowerHaveTwoCommonVertices *(Рис.36)*

Тест проверяет, клумбы с двумя общими вершинами

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=5, y1d=5, x2d=15, y2d=15

**Ожидаемый результат:** 225

*Рис.36.* TestFlowerHaveTwoCommonVertices

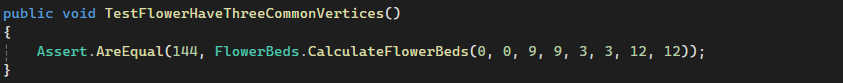
13 тест: TestFlowerHaveThreeCommonVertices *(Рис.37)*

Тест проверяет, клумбы с тремя общими вершинами

**Входные данные:**

x1m=0, y1m=0, x2m=9, y2m=9, x1d=3, y1d=3, x2d=12, y2d=12

**Ожидаемый результат:** 144

*Рис.37.* TestFlowerHaveThreeCommonVertices

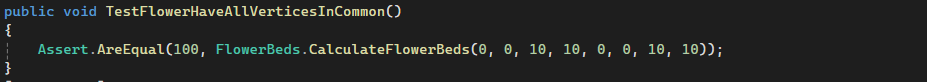
14 тест: TestFlowerHaveAllVerticesInCommon *(Рис.38)*

Тест проверяет, клумбы со всеми четырьмя общими вершинами

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=0, y1d=0, x2d=10, y2d=10

**Ожидаемый результат:** 100

*Рис.38.* TestFlowerHaveAllVerticesInCommon

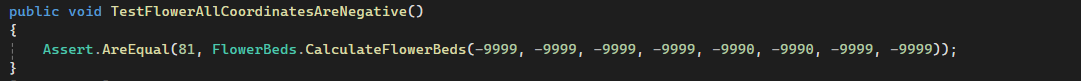
15 тест: TestFlowerAllCoordinatesAreNegative *(Рис.39)*

Тест проверяет, все координаты отрицательные

**Входные данные:**

x1m=-9999, y1m=-9999, x2m=-9999, y2m=-9990, x1d=-9990, y1d=-9990, x2d=-9999, y2d=-9999

**Ожидаемый результат:** 81

*Рис.39.* TestFlowerAllCoordinatesAreNegative

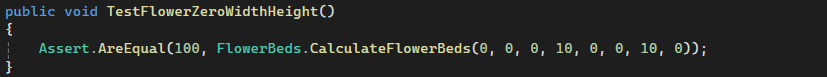
16 тест: TestFlowerZeroWidthHeight *(Рис.40)*

Тест проверяет, цветочные клумбы с нулевой шириной или высотой

**Входные данные:**

x1m=0, y1m=0, x2m=0, y2m=10, x1d=0, y1d=0, x2d=10, y2d=0

**Ожидаемый результат:** 100

*Рис.40.* TestFlowerZeroWidthHeight

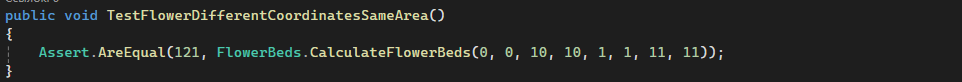
17 тест: TestFlowerDifferentCoordinatesSameArea *(Рис.41)*

Тест проверяет, что клумбы имеют одинаковую площадь, но разные координаты

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=1, y1d=1, x2d=11, y2d=11

**Ожидаемый результат:** 121

*Рис.41.* TestFlowerDifferentCoordinatesSameArea

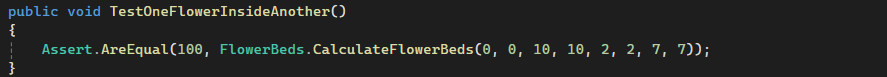
18 тест: TestOneFlowerInsideAnother *(Рис.42)*

Тест проверяет, что одна клумба полностью внутри другой

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=2, y1d=2, x2d=7, y2d=7

**Ожидаемый результат:** 100

*Рис.42.* TestOneFlowerInsideAnother

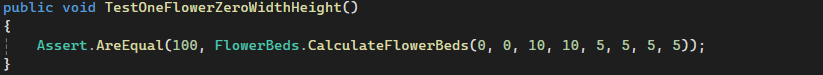
19 тест: TestOneFlowerZeroWidthHeight *(Рис.43)*

Тест проверяет, что клумбы имеют нулевую ширину и высоту, но одна из них не нулевая

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=5, y1d=5, x2d=5, y2d=5

**Ожидаемый результат:** 100

*Рис.43.* TestOneFlowerZeroWidthHeight

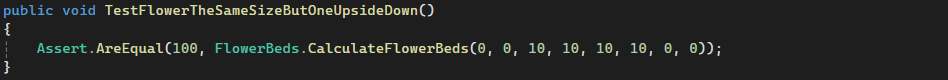
20 тест: TestFlowerTheSameSizeButOneUpsideDown *(Рис.44)*

Тест проверяет, что клумбы имеют одинаковый размер, но одна перевернута

**Входные данные:**

x1m=0, y1m=0, x2m=10, y2m=10, x1d=10, y1d=10, x2d=0, y2d=0

**Ожидаемый результат:**100

*Рис.44.* TestFlowerTheSameSizeButOneUpsideDown

## 1.6. Используемые инструменты

**Вариант 1 и вариант 3:**

## Язык написания данной программы- C#

## Среда разработки – WindowsForms C#

## 1.7. Описание пользовательского интерфейса

**Вариант 1:**

Приведённый код вычисляет стоимость телефонного звонка на основе нескольких параметров. Он считывает входные данные из файла, обрабатывает их и записывает результат в выходной файл. В этом проекте реализованы модульные тесты для проверки функциональности метода Calculate класса Connection, который рассчитывает стоимость соединения на основе заданных параметров.

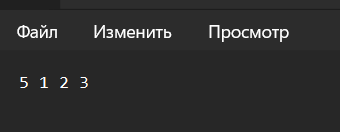
**Вариант 3:**

Приведённый код, позволяет считывать данные о координатах двух прямоугольников (клумб для мальчиков и девочек) из файла input.txt, вычислять их площади и площадь пересечения, а затем записывать результат в файл output.txt. В этом проекте реализованы модульные тесты, предназначенных для проверки функциональности метода CalculateFlowerBeds из класса FlowerBeds. Эти тесты для различных ситуаций, связанные с пересечением и свойствами клумб, представленных в виде прямоугольников.

## 1.8. Результат

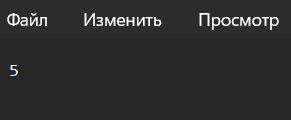
**Вариант 1:**

Файл ввода информации *(Рис.45):*



*Рис.45. Input*

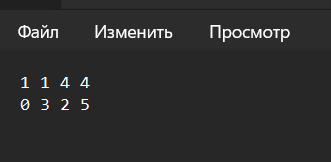
Файл вывода информации *(Рис.46):*



*Рис.46. Output*

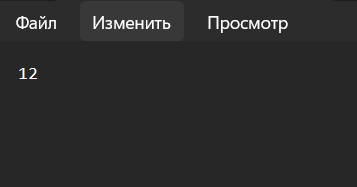
**Вариант 3:**

Файл ввода информации *(Рис.47):*



*Рис.47. Input*

Файл вывода информации *(Рис.48):*



*Рис.48. Input*