

Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт космических и информационных технологий  
Кафедра информатики

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1**

по введению в профессиональную деятельность

тема: Работа с командной строкой

Вариант 10

Преподаватель

П.В. Пересунько

Студент ЗКИ21-16Б № з\к 031729540

Д.А. Павлова

Красноярск 2023

## СОДЕРЖАНИЕ

Введение.....	3
1 Описание варианта задания.....	4
2 Ход выполнения работы.....	4
Заключение.....	9
Список использованных источников.....	10

## **ВВЕДЕНИЕ**

Данная практическая работа преследует следующие цели: ознакомление с командной строкой (CMD) и CLI, использование основных команд для выполнения практических заданий. Также работа преследует цель научиться создавать скрипты для автоматизации деятельности и создавать и запускать программы с аргументами командной строки.

Также данная практическая работа имеет задачи в виде изучения основных принципов работы с CMD, приобретение навыков по реализации пакетного файла (BAT), который выполняет команды для CMD, написании программы, реализующую интерфейс командной строки (Command Line Interface или CLI).

Необходимо также составить отчет о проделанной практической работе в соответствии со стандартами организации.

## 1 Описание варианта задания

В данном 10 варианте необходимо написать программу, которая будет проводить расчет площади треугольника по трем сторонам. Для данного расчета используется следующая формула:

$$S = \sqrt{p * (p - a) * (p - b) * (p - c)}, \quad (1)$$

где  $S$  – площадь треугольника;

$p$  – полупериметр;

$a, b, c$  – стороны треугольника

Программа должна быть написана на программном языке Python. Затем, нужно написать отчет о проделанной работе, предоставить программу и отчет на проверку, ответить на вопросы и выполнить дополнительное задание.

## 2 Ход выполнения работы

Сначала программа должна корректно принимать аргументы через командную строку. Для этого была использована библиотека `argparse`. Данная библиотека помогает анализировать переданные аргументы, назначать им различные свойства и проверять на соответствие им.

Тип аргумента у нас только один – это сторона треугольника. Однако, их нужно определенное количество – именно 3, ни больше, ни меньше.

Для начала создадим объект `argparser`'а, добавим описание для нашей программы (что она делает), а затем добавим для него новый аргумент в виде стороны треугольника (рисунок 1).

```
1 import math
2 import argparse
3
4 parser = argparse.ArgumentParser(description="Program calculates the square "
5 | | | | | | | | | | + "of a triangle.")
6 parser.add_argument('side', type=float, help="Length of a side", nargs=3)
7 args = parser.parse_args()
8 sides = args.side
```

Рисунок 1 – Создание парсера и добавление аргумента

Как видно из переданных аргументов при создании нашего нового аргумента (рисунок 1, строка 6), он будет иметь название “side”, будет типа “float”. Help задает описание к аргументу, которое объясняет, что это длина стороны. Nargs=3 означает, что данных аргументов должно быть передано в программу именно 3 штуки. Вызываем функцию parse\_args() для парсирования переданных аргументов и их анализа, затем сохраняем аргументы в переменную sides для более удобного обращения к ним.

Если аргументы были переданы верные (3 аргумента типа флот), то далее начинает работу наша программа. Её основная функция выглядит весьма лаконично (рисунок 2).

```
11 def main():
12     if is_valid_triangle() is False:
13         return
14     square_triangle = calculate_square_triangle()
15     print("Square of the triangle is " + str(round(square_triangle, 2)))
```

Рисунок 2 – Основная функция программы.

Для начала эта функция проверяет, были ли переданы корректные данные. Для этого вызывается другая функция `is_valid_triangle()`, которая делает проверку и возвращает булево значение `true` или `false`. Код этой функции показан на рисунке 3.

```

32 def is_valid_triangle():
33     for side in sides:
34         if side <= 0:
35             print("Side's length must be greater than zero!")
36             return False
37     side_a = sides[0]
38     side_b = sides[1]
39     side_c = sides[2]
40     if (side_a >= side_b + side_c) or (side_b >= side_a + side_c) \
41         or (side_c >= side_a + side_b):
42         print("Such triangle does not exist! Any side's length cannot\n"
43             + "exceed the sum of the two other sides!")
44         return False
45     return True

```

Рисунок 3 – Код проверки данных для треугольника

Для начала функция смотрит, чтобы значения сторон не были отрицательными (рисунок 3, строки 33-36). Далее извлекает из списка стороны и заносит их в переменные, после чего производит расчет того, что треугольник с такими сторонами действительно может существовать.

Дело в том, что если одна из сторон треугольника будет больше, чем сумма двух других сторон, то такого треугольника не может существовать в природе. Поэтому функция делает данную проверку и возвращает false, если хотя бы одна из сторон больше суммы двух остальных (рисунок 3, строки 40-44).

Если же обе проверки прошли успешно, то функция возвращает true.

Для вычисления площади треугольника нужно найти его периметр для формулы (1). Этим занимается функция calculate\_perimeter() (рисунок 4).

```

25 def calculate_perimeter():
26     perimeter = 0
27     for side in sides:
28         perimeter += side
29     return perimeter

```

Рисунок 4 – Код для вычисления периметра треугольника

Функция берет наш список сторон и последовательно прибавляет их к переменной «perimeter» с изначальным значением в 0, затем возвращает эту переменную.

Функция по вычислению площади треугольника изображена на рисунке 5.

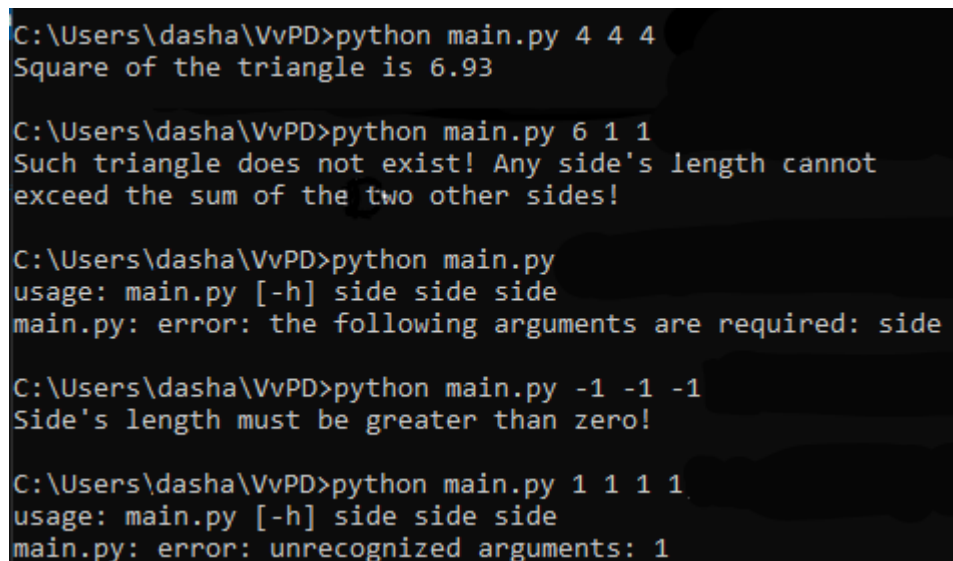
```
18 def calculate_square_triangle():
19     perimeter = calculate_perimeter()
20     p = perimeter / 2 # half of the perimeter for the formula
21     square = math.sqrt(p * (p - sides[0]) * (p - sides[1]) * (p - sides[2]))
22     return square
--
```

Рисунок 5 – Вычисление площади треугольника

Сначала она запускает вышеописанную функцию по вычислению периметра, затем делит периметр на два (т.к. для формулы (1) нужен полупериметр), а затем реализует формулу (1) и возвращает значение.

После чего основная функция программы выводит значение и завершает работу.

Демонстрация работы программы показан на рисунке 6



```
C:\Users\dasha\VvPD>python main.py 4 4 4
Square of the triangle is 6.93

C:\Users\dasha\VvPD>python main.py 6 1 1
Such triangle does not exist! Any side's length cannot
exceed the sum of the two other sides!

C:\Users\dasha\VvPD>python main.py
usage: main.py [-h] side side side
main.py: error: the following arguments are required: side

C:\Users\dasha\VvPD>python main.py -1 -1 -1
Side's length must be greater than zero!

C:\Users\dasha\VvPD>python main.py 1 1 1 1
usage: main.py [-h] side side side
main.py: error: unrecognized arguments: 1
```

Рисунок 6 – Демонстрация работы программы

Сначала программа получает корректные аргументы и вычисляет площадь треугольника, выводит ее на экран. В последующие запуски передаются неверные аргументы и показана обработка таких случаев программой. А именно: для несуществующего треугольника, без аргументов, с отрицательными сторонами и с 4 аргументами.



## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы были приобретены навыки с работой с Command Line Interface, осуществлено знакомство с bat скриптами. Также были углублены знания языка программирования Python. Более подробно были изучены требования к оформлению работ, описанные в стандарте организации СФУ.

Как итог, в ходе выполнения данной практической работы были достигнуты все поставленные цели и задачи, написана программа для вычисления площади треугольника по трем сторонам.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТО СФУ 2021. Стандарт университета. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. Номер документа СТУ 7.5-07-2021, дата принятия 2021 г. URL: <https://about.sfu-kras.ru/docs/8127/pdf/456858> (дата обращения 01.06.2023).

2. Официальная документация к языку программирования Python. Python.org: официальный сайт. URL: <https://docs.python.org/3/tutorial/index.html> (дата обращения 01.06.2023).