Отчет по третьей практической работе

В ходе выполнения первого задания получилась программа со следующим кодом:

def days\_lived(year\_of\_birth):

from datetime import datetime

current\_year = datetime.now().year

current\_age = current\_year - year\_of\_birth

days\_lived = 0

for year in range(year\_of\_birth, current\_year + 1):

if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:

days\_lived += 366

else:

days\_lived += 365

return days\_lived, current\_age

if \_\_name\_\_ == "\_\_main\_\_":

year\_of\_birth = int(input("Введите год рождения: "))

total\_days\_lived, current\_age = days\_lived(year\_of\_birth)

print(f"Прожито дней: {total\_days\_lived}")

print(f"Полный возраст: {current\_age}")

Проверка работоспособности:



Рисунок 1 Количество прожитых дней

В ходе выполнения второго задания получилась программа со следующим кодом:

def count\_substring\_occurrences(substring, main\_string):

count = 0

start = 0

while True:

start = main\_string.find(substring, start)

if start == -1:

break

count += 1

start += 1

return count

main\_string = input("Введите строку для поиска подстроки: ")

substring = input("Введите подстроку: ")

result = count\_substring\_occurrences(substring, main\_string)

print(f"Количество встреченных подстрок: {result}")

Проверка работоспособности:

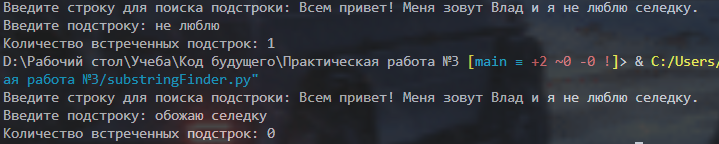


Рисунок 2 Строка в подстроке

В ходе выполнения третьего задания получилась программа со следующим кодом:

import math

def calculate\_discriminant(a, b, c):

return b\*\*2 - 4\*a\*c

def calculate\_roots(a, b, c):

d = calculate\_discriminant(a, b, c)

if d > 0:

x1 = (-b + math.sqrt(d)) / (2\*a)

x2 = (-b - math.sqrt(d)) / (2\*a)

return x1, x2

elif d == 0:

x = -b / (2\*a)

return x

else:

return "Нет "

def calculate\_probability(m, n):

return m / n

def calculate\_sum\_of\_independent\_probabilities(pa, pb):

return pa + pb

def calculate\_sum\_of\_joint\_probabilities(pa, pb, pab):

return pa + pb - pab

В ходе выполнения четвертого задания получилась программа со следующим кодом:

class Engine:

def \_\_init\_\_(self, weight, max\_speed, fuel\_consumption):

self.weight = weight

self.max\_speed = max\_speed

self.fuel\_consumption = fuel\_consumption

class Tank:

def \_\_init\_\_(self, weight, tank\_volume):

self.weight = weight

self.tank\_volume = tank\_volume

class Brakes:

def \_\_init\_\_(self, weight, brake\_efficiency):

self.weight = weight

self.brake\_efficiency = brake\_efficiency

class Body:

def \_\_init\_\_(self, weight):

self.weight = weight

class Car:

def \_\_init\_\_(self, name, engine, tank, brakes, body):

self.name = name

self.engine = engine

self.tank = tank

self.brakes = brakes

self.body = body

self.weight = self.engine.weight + self.tank.weight + self.brakes.weight + self.body.weight

self.max\_distance\_per\_tank = (self.tank.tank\_volume / self.engine.fuel\_consumption) \* self.engine.max\_speed

self.braking\_distance = self.weight / self.brakes.brake\_efficiency

if \_\_name\_\_ == "\_\_main\_\_":

engine\_weight = float(input("Введите вес двигателя: "))

engine\_max\_speed = float(input("Введите максимальную скорость двигателя: "))

engine\_fuel\_consumption = float(input("Введите потребление топлива двигателем: "))

tank\_weight = float(input("Введите вес бака: "))

tank\_volume = float(input("Введите объем бака: "))

brakes\_weight = float(input("Введите вес тормозов: "))

brake\_efficiency = float(input("Введите эффективность тормозов: "))

body\_weight = float(input("Введите вес кузова: "))

engine = Engine(engine\_weight, engine\_max\_speed, engine\_fuel\_consumption)

tank = Tank(tank\_weight, tank\_volume)

brakes = Brakes(brakes\_weight, brake\_efficiency)

body = Body(body\_weight)

car\_name = input("Введите название машины: ")

car = Car(car\_name, engine, tank, brakes, body)

print(f"Название машины: {car.name}")

print(f"Вес машины: {car.weight} кг")

print(f"Максимальный пробег на одном баке: {car.max\_distance\_per\_tank} км")

print(f"Тормозной путь: {car.braking\_distance} метров")

Проверка работоспособности:

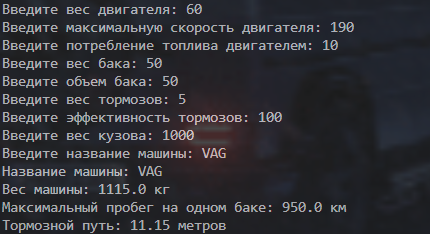


Рисунок 3 Расчет характеристик