Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

"Российский экономический университет имени Г. В. Плеханова"

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность 09.02.07 Информационные системы и программирование

Квалификация: Программист

ОТЧЕТ

КОД БУДУЩЕГО

Листов: 18

|  |  |
| --- | --- |
| Выполнила студент | Проверил преподаватель |
| Группы П50-5-20 | \_\_\_\_\_\_\_\_\_\_Д.В. Серяк |
| Завьялов М.А. | " " \_\_\_\_\_\_\_\_2023 года |

Москва 2023

# Практическая работа №3

## Тема «Функции, модули, ООП».

Цель работы: изучить работу функций, модулей и ООП.

Ход работы:

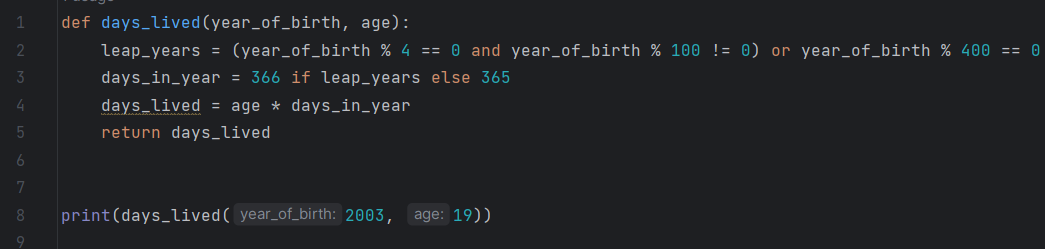


Рисунок 1 – Код расчёта возраста

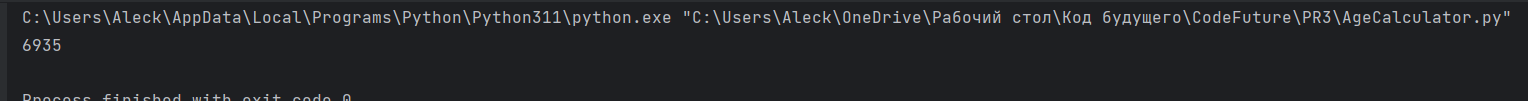


Рисунок 2 – Результат работы программы

AgeCalculator.py

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

leap\_years = (year\_of\_birth % 4 == 0 and year\_of\_birth % 100 != 0) or year\_of\_birth % 400 == 0

days\_in\_year = 366 if leap\_years else 365

days\_lived = age \* days\_in\_year

return days\_lived

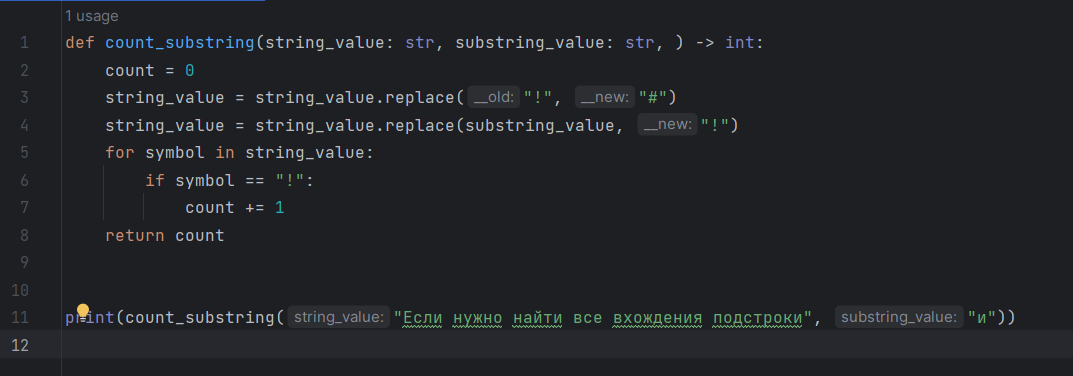


Рисунок 3 – Количество подстрок в строке

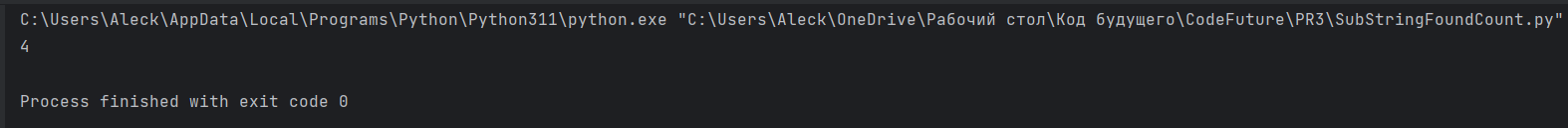


Рисунок 4 – Вывод подстрок

print(days\_lived(2003, 19))

SubStringFoundCount.py

def count\_substring(string\_value: str, substring\_value: str, ) -> int:

count = 0

string\_value = string\_value.replace("!", "#")

string\_value = string\_value.replace(substring\_value, "!")

for symbol in string\_value:

if symbol == "!":

count += 1

return count

print(count\_substring("Если нужно найти все вхождения подстроки", "и"))

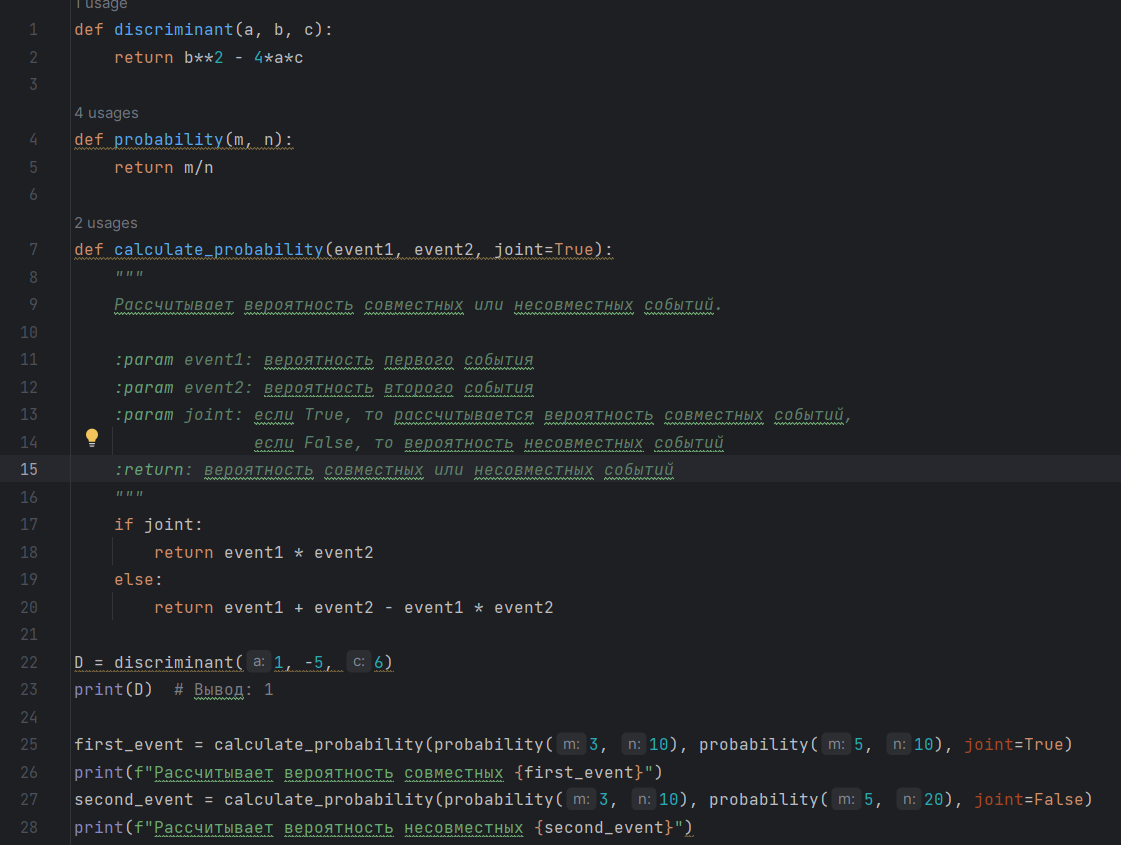


Рисунок 5 – Код программы модули

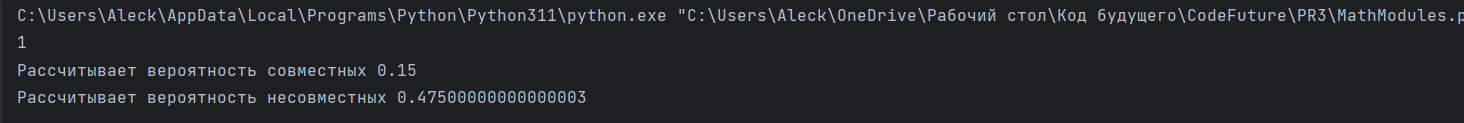


Рисунок 6 – Расчёт дискриминантов, вероятных событий и невероятных событий

MathModules.py

def discriminant(a, b, c):

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

def probability(m, n):

return m/n

def calculate\_probability(event1, event2, joint=True):

"""

Рассчитывает вероятность совместных или несовместных событий.

:param event1: вероятность первого события

:param event2: вероятность второго события

:param joint: если True, то рассчитывается вероятность совместных событий,

если False, то вероятность несовместных событий

:return: вероятность совместных или несовместных событий

"""

if joint:

return event1 \* event2

else:

return event1 + event2 - event1 \* event2

D = discriminant(1, -5, 6)

print(D) # Вывод: 1

first\_event = calculate\_probability(probability(3, 10), probability(5, 10), joint=True)

print(f"Рассчитывает вероятность совместных {first\_event}")

second\_event = calculate\_probability(probability(3, 10), probability(5, 20), joint=False)

print(f"Рассчитывает вероятность несовместных {second\_event}")

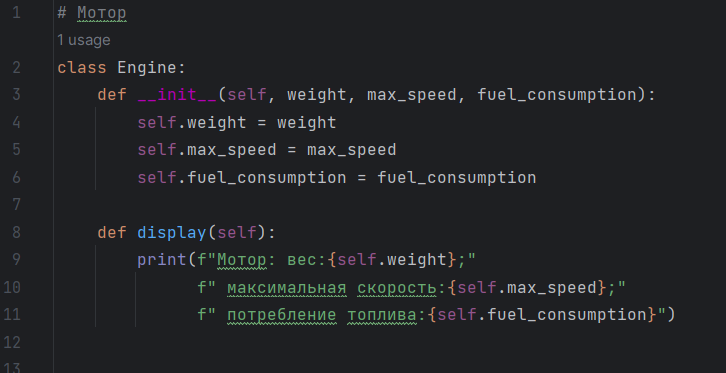


Рисунок 7 – Класс мотор

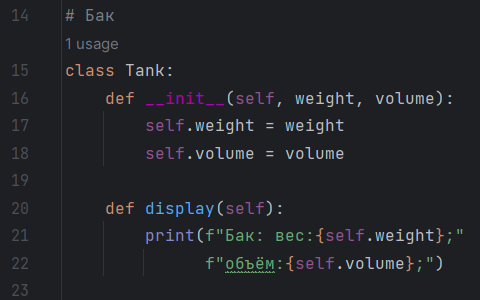


Рисунок 8 – Класс бак

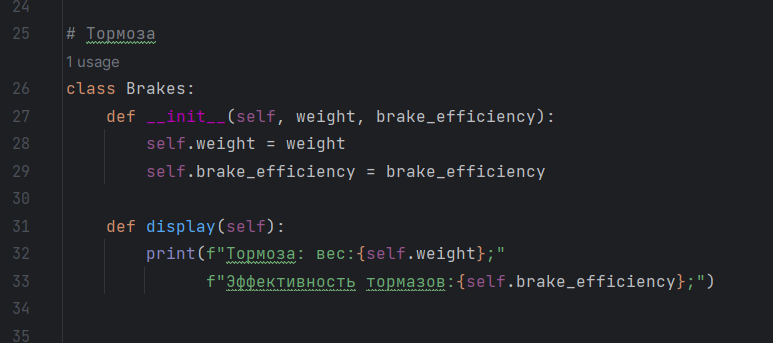


Рисунок 9 – Класс тормоза

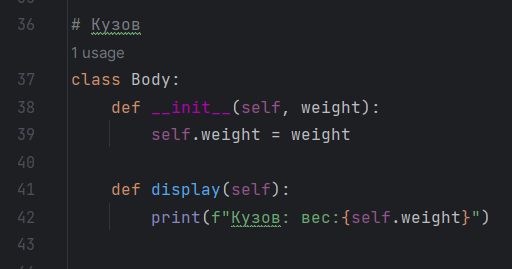


Рисунок 10 – Класс кузов

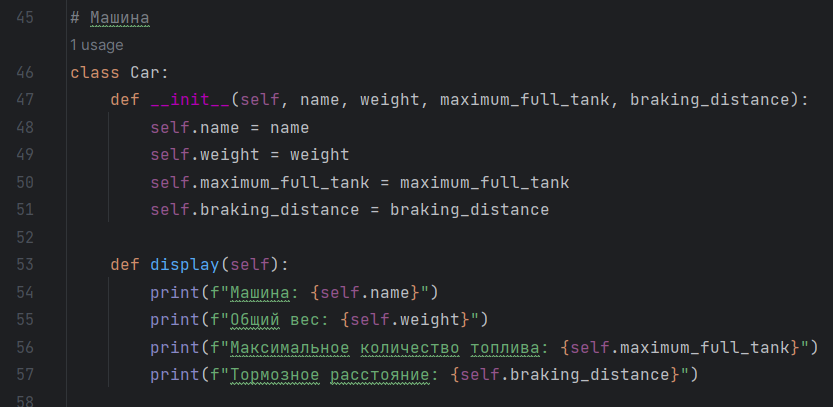


Рисунок 11 – Класс машина

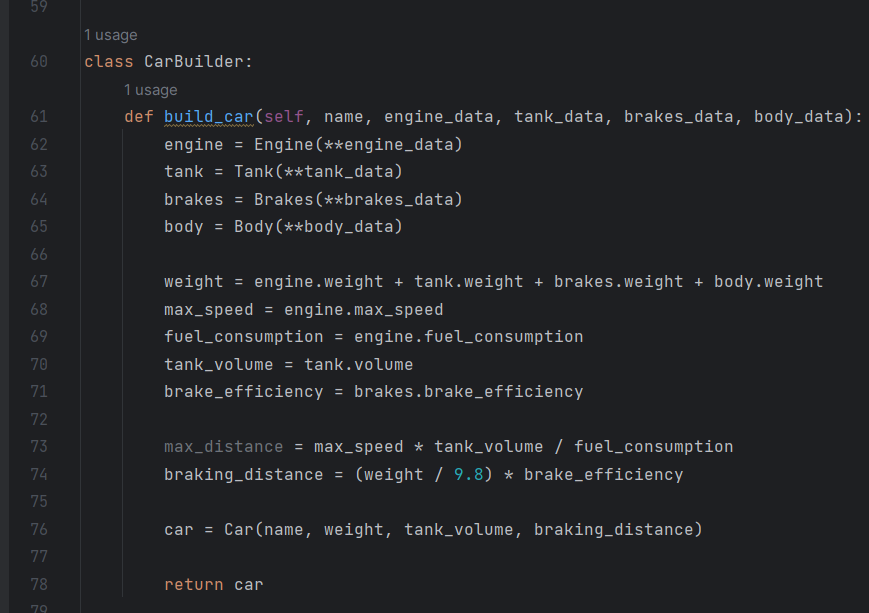


Рисунок 12 – Класс сборщик

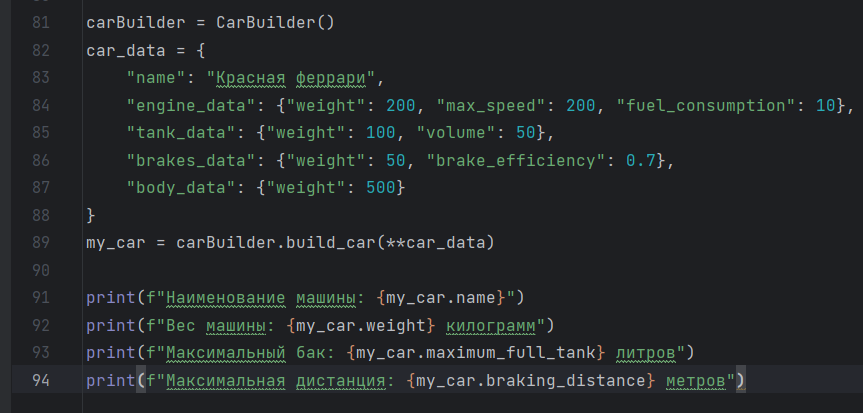


Рисунок 13 – Вывод машины

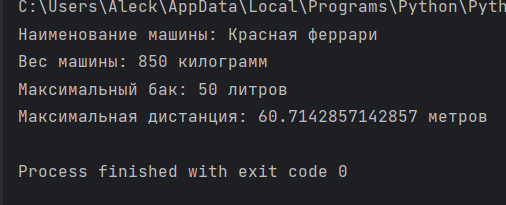


Рисунок 14 – Результат

Collector.py

# Мотор

class Engine:

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

self.weight = weight

self.max\_speed = max\_speed

self.fuel\_consumption = fuel\_consumption

def display(self):

print(f"Мотор: вес:{self.weight};"

f" максимальная скорость:{self.max\_speed};"

f" потребление топлива:{self.fuel\_consumption}")

# Бак

class Tank:

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

self.weight = weight

self.volume = volume

def display(self):

print(f"Бак: вес:{self.weight};"

f"объём:{self.volume};")

# Тормоза

class Brakes:

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

self.weight = weight

self.brake\_efficiency = brake\_efficiency

def display(self):

print(f"Тормоза: вес:{self.weight};"

f"Эффективность тормазов:{self.brake\_efficiency};")

# Кузов

class Body:

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

self.weight = weight

def display(self):

print(f"Кузов: вес:{self.weight}")

# Машина

class Car:

def \_\_init\_\_(self, name, weight, maximum\_full\_tank, braking\_distance):

self.name = name

self.weight = weight

self.maximum\_full\_tank = maximum\_full\_tank

self.braking\_distance = braking\_distance

def display(self):

print(f"Машина: {self.name}")

print(f"Общий вес: {self.weight}")

print(f"Максимальное количество топлива: {self.maximum\_full\_tank}")

print(f"Тормозное расстояние: {self.braking\_distance}")

class CarBuilder:

def build\_car(self, name, engine\_data, tank\_data, brakes\_data, body\_data):

engine = Engine(\*\*engine\_data)

tank = Tank(\*\*tank\_data)

brakes = Brakes(\*\*brakes\_data)

body = Body(\*\*body\_data)

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

max\_speed = engine.max\_speed

fuel\_consumption = engine.fuel\_consumption

tank\_volume = tank.volume

brake\_efficiency = brakes.brake\_efficiency

max\_distance = max\_speed \* tank\_volume / fuel\_consumption

braking\_distance = (weight / 9.8) \* brake\_efficiency

car = Car(name, weight, tank\_volume, braking\_distance)

return car

carBuilder = CarBuilder()

car\_data = {

"name": "Красная феррари",

"engine\_data": {"weight": 200, "max\_speed": 200, "fuel\_consumption": 10},

"tank\_data": {"weight": 100, "volume": 50},

"brakes\_data": {"weight": 50, "brake\_efficiency": 0.7},

"body\_data": {"weight": 500}

}

my\_car = carBuilder.build\_car(\*\*car\_data)

print(f"Наименование машины: {my\_car.name}")

print(f"Вес машины: {my\_car.weight} килограмм")

print(f"Максимальный бак: {my\_car.maximum\_full\_tank} литров")

print(f"Максимальная дистанция: {my\_car.braking\_distance} метров")

Вывод: изучил работу функций, модулей и ООП.

# Список иллюстраций

[Рисунок 1 – Код расчёта возраста 2](#_Toc149503052)

[Рисунок 2 – Результат работы программы 2](#_Toc149503053)

[Рисунок 3 – Количество подстрок в строке 2](#_Toc149503054)

[Рисунок 4 – Вывод подстрок 2](#_Toc149503055)

[Рисунок 5 – Код программы модули 3](#_Toc149503056)

[Рисунок 6 – Расчёт дискриминантов, вероятных событий и невероятных событий 3](#_Toc149503057)

[Рисунок 7 – Класс мотор 4](#_Toc149503058)

[Рисунок 8 – Класс бак 5](#_Toc149503059)

[Рисунок 9 – Класс тормоза 5](#_Toc149503060)

[Рисунок 10 – Класс кузов 6](#_Toc149503061)

[Рисунок 11 – Класс машина 6](#_Toc149503062)

[Рисунок 12 – Класс сборщик 7](#_Toc149503063)

[Рисунок 13 – Вывод машины 7](#_Toc149503064)

[Рисунок 14 – Результат 8](#_Toc149503065)