

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

**Отчет по лабораторной работе №4
«Основные конструкции языка Python»**

**Выполнил:
студент группы ИУ5-
32Б
Давиташвили Шако**

Москва, 2021 г.

Описание задания:

- 1) Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.

Текст программы, реализация паттерна «Фасад»:

Файл pattern.py:

```
class _IgnitionSystem(object):

    @staticmethod
    def produce_spark():
        return True

class _Engine(object):

    def __init__(self):
        self.revs_per_minute = 0

    def turnon(self):
        self.revs_per_minute = 2000

    def turnoff(self):
        self.revs_per_minute = 0

class _FuelTank(object):

    def __init__(self, level=30):
        self._level = level

    @property
    def level(self):
        return self._level

    @level.setter
    def level(self, level):

        self._level = level

class _DashBoardLight(object):

    def __init__(self, is_on=False):
        self._is_on = is_on

    def __str__(self):
        return self.__class__.__name__

    @property
    def is_on(self):
        return self._is_on

    @is_on.setter
    def is_on(self, status):
        self._is_on = status
```

```

def status_check(self):
    if self._is_on:
        print("{}: ON".format(str(self)))
    else:
        print("{}: OFF".format(str(self)))

class _HandBrakeLight(_DashBoardLight):
    pass

class _FogLampLight(_DashBoardLight):
    pass

class _Dashboard(object):

    def __init__(self):
        self.lights = {"handbreak": _HandBrakeLight(), "fog": _FogLampLight()}

    def show(self):
        for light in self.lights.values():
            light.status_check()

class Fasade(object):

    def __init__(self):
        self.ignition_system = _IgnitionSystem()
        self.engine = _Engine()
        self.fuel_tank = _FuelTank()
        self.dashboard = _Dashboard()

    @property
    def km_per_litre(self):
        return 17.0

    def consume_fuel(self, km):
        litres = min(self.fuel_tank.level, km / self.km_per_litre)
        self.fuel_tank.level -= litres

    def start(self):
        print("\nStarting...")
        self.dashboard.show()
        if self.ignition_system.produce_spark():
            self.engine.turnon()
        else:
            print("Can't start. Faulty ignition system")

    def has_enough_fuel(self, km, km_per_litre):
        litres_needed = km / km_per_litre
        if self.fuel_tank.level > litres_needed:
            return True
        else:
            return False

    def drive(self, km = 100):
        print("\n")
        if self.engine.revs_per_minute > 0:
            while self.has_enough_fuel(km, self.km_per_litre):

```

```

        self.consume_fuel(km)
        print("Drove {}km".format(km))
        print("{:.2f}l of fuel still left".format(self.fuel_tank.level))
    else:
        print("Can't drive. The Engine is turned off!")

def park(self):
    print("\nParking...")
    self.dashboard.lights["handbreak"].is_on = True
    self.dashboard.show()
    self.engine.turnoff()

def switch_fog_lights(self, status):
    print("\nSwitching {} fog lights...".format(status))
    boolean = True if status == "ON" else False
    self.dashboard.lights["fog"].is_on = boolean
    self.dashboard.show()

def fill_up_tank(self):
    print("\nFuel tank filled up!")
    self.fuel_tank.level = 100

def main():
    car = Facade()
    car.start()
    car.drive()
    car.switch_fog_lights("ON")
    car.switch_fog_lights("OFF")
    car.park()
    car.fill_up_tank()
    car.drive()
    car.start()
    car.drive()

if __name__ == "__main__":
    main()

```

Пример выполнения программы:

Starting...

_HandBrakeLight: ON

_FogLampLight: OFF

Drove 100km

94.12l of fuel still left

Drove 100km

88.24l of fuel still left

Drove 100km

82.35l of fuel still left

Drove 100km

76.47l of fuel still left

Drove 100km

70.59l of fuel still left

Drove 100km

64.71l of fuel still left

Drove 100km

58.82l of fuel still left

Drove 100km

52.94l of fuel still left

Drove 100km

47.06l of fuel still left

Drove 100km

41.18l of fuel still left

Drove 100km

35.29l of fuel still left

Drove 100km

29.41l of fuel still left

Drove 100km

23.53l of fuel still left

Drove 100km

17.65l of fuel still left

Drove 100km

11.76l of fuel still left

Drove 100km

5.88l of fuel still left

Drove 100km

0.00l of fuel still left