

**Министерство образования Республики Беларусь**

**Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»**

---

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ**

**Кафедра интеллектуальных информационных технологий**

**Отчет по лабораторной работе №1**

Выполнил: Протас А.А., 121701

Проверил: Бутрин С. В.

**Минск 2023**

## Описание класса Station

```
import enum
from enum import Enum

class StationType(enum.IntEnum):
    PASS_FREIGHT_STATION = 1
    PASS_STATION = 2
    FREIGHT_STATION = 3

class Station:
    def __init__(self, stationName, stationType, distanceToNextStation):
        self.stationName = stationName
        self.stationType = stationType
        self.distanceToNextStation = distanceToNextStation

    def setDistanceToNextStation(self, distanceToNextStation):
        self.distanceToNextStation = distanceToNextStation

    def getStationType(self):
        return self.stationType

    def getStationName(self):
        return self.stationName

    def getDistanceToNextStation(self):
        return self.distanceToNextStation
```

## Описание класса Railway

```
import Train

class Railway:
    def __init__(self, stations, train: Train):
        self.stations = stations
        self.train = train

    def start_train(self):
        size = len(self.stations)
        for i in range(size):
            if i == size - 1:
                current_station = self.stations[i]
                self.train.processStation(current_station, True)
            else:
                current_station = self.stations[i]
                self.train.processStation(current_station, False)
```

## Описание модели поезда

```
import Randomizer
import time

class Train:
    MAX_PASS_BY_CAR = 50
    MAX_MASS_BY_CAR = 68
    SPEED_LOST_BY_MASS = 0.005
    SPEED_LOST_BY_PASSENGERS = 0.001

    def __init__(self, default_train_speed, pass_cars, freight_cars):
        self.default_train_speed = default_train_speed
        self.pass_cars = pass_cars
        self.freight_cars = freight_cars
        self.actual_pass = 0
        self.actual_mass = 0
        self.actual_train_speed = default_train_speed

    def processPassengers(self, station):
        randomazer = Randomizer.Randomazer()
        passengersLeave = randomazer.generateRandomValue(0, self.actual_pass)
        print(f"Train lose {passengersLeave} passengers")
        self.actual_pass -= passengersLeave
        passengersGet = randomazer.generateRandomValue(0, (self.MAX_PASS_BY_CAR * self.pass_cars) - self.actual_pass)
        print(f"Train get {passengersGet} passengers")
        self.actual_pass += passengersGet
        print(f"- In train now {self.actual_pass} passengers")

    def processFreight(self, station):
        randomazer = Randomizer.Randomazer()
        massLeave = randomazer.generateRandomValue(0, self.actual_mass)
        print(f"Train lose {massLeave} tons")
        self.actual_mass -= massLeave
        massGet = randomazer.generateRandomValue(0, (self.MAX_MASS_BY_CAR * self.freight_cars) - self.actual_mass)
        print(f"Train get {massGet} tons")
        self.actual_mass += massGet
        print(f"- In train now {self.actual_mass} tons")

    def calculateSpeed(self, station):
        self.actual_train_speed -= (self.actual_mass * self.SPEED_LOST_BY_MASS) + (
            self.actual_pass * self.SPEED_LOST_BY_PASSENGERS)
        print(f"Now train is moving {self.actual_train_speed} km/h")
        print(f"Distance to the next station {station.getDistanceToNextStation()} kilometers")
        times = station.getDistanceToNextStation() / self.actual_train_speed
        print(f"Train will arrive to the next station after {times} hours")
        time.sleep(times * 10)

    def processStation(self, station, isLast):
        stationType = station.getStationType()
        print("\t\tTrain arrive to " + station.getStationName())

        if not isLast:
            stationType1 = station.getStationType()
            if stationType1 == 1:
                self.processPassengers(station)
                self.processFreight(station)
            elif stationType1 == 2:
                self.processPassengers(station)
            elif stationType1 == 3:
                self.processFreight(station)

            self.calculateSpeed(station)
```

```

else:
    print("Train lose " + str(self.actual_pass) + " passengers")
    self.actual_pass -= self.actual_pass
    print("Train get 0 passengers")
    print("- In train now " + str(self.actual_pass) + " passengers")

    print("Train lose " + str(self.actual_mass) + " tons")
    self.actual_mass -= self.actual_mass
    print("Train get 0 tons")
    print("- In train now " + str(self.actual_mass) + " tons")
    print("\t\t\tTrain is STOPPED!!!")

print("=====")

```

```

class Train:

```

```

    def __init__(self):
        pass_cars = int(input("Passenger cars - "))
        freight_cars = int(input("Freight cars - "))
        default_train_speed = float(input("Default speed - "))
        self.pass_cars = pass_cars
        self.freight_cars = freight_cars
        self.default_train_speed = default_train_speed
        self.actual_train_speed = default_train_speed

```

## Описание класса Randomazer

```
import random

import Station

class Randomazer:

    def generateRandomValue(self, min, max):
        num = random.randint(min, max)
        return num

    def generateRandomStation(self):
        stations = []

        for i in range(10):
            station = Station.Station(f"Station_{i + 1}", random.randint(1, 3), self.generateRandomValue(30, 100))
            stations.append(station)
        return stations
```

## Запуск программы с помощью CLI

```
import random
import Randomizer
import Railway
import Train
import click

@click.group()
def cli():
    pass

@click.command()
@click.argument('train-speed')
@click.argument('pass-cars')
@click.argument('freight-cars')
def start(train_speed, pass_cars, freight_cars):
    random.seed()
    train = Train.Train(int(train_speed), int(pass_cars), int(freight_cars))
    randomazerr = Randomizer.Randomizer()
    railway = Railway.Railway(randomazerr.generateRandomStation(), train)
    railway.start_train()

cli.add_command(start)

if __name__ == '__main__':
    cli()
```