

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського» Факультет
інформатики та обчислювальної техніки Кафедра
інформатики та програмної інженерії

Звіт

З лабораторної роботи №3 з дисципліни

«Основи програмування 2.

Модульне програмування»

«Класи та об'єкти»

Варіант 27

Виконав студент

ІП-11 Савенко Олексій Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Вітковська І.І.

(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 3

Класи та об'єкти

Мета роботи – вивчити механізми створення та використання класів та об'єктів

Варіант 27

27. Розробити клас "Поїзд", який характеризується номером поїзда, пунктом призначення, часом його відправлення (у форматі ГГ-ХХ). Створити масив об'єктів даного класу. Визначити поїзди (їх номери), які відправляються з вокзалу у заданий період часу (діапазон годин).

Постановка завдання

Створити клас, який імітує залізничний потяг(повинен містити такі атрибути потяга як час, номер потягу, а також напрям його поїздки), потрібно реалізувати один з принципів ООП – інкапсуляцію. Реалізувати введення даних, щодо характеристик потягу від користувача, утворивши масив потягів для подальшої обробки цих об'єктів, а також отримання від нього ж часового проміжку для визначення чи входить час відправлення утворених потягів до цього проміжку. Після цього потрібно написати функціонал для перевірки потягів на належність до часового проміжку відправлення потягів.

Програмний код на мові C#:

Program.cs:

```
// суть ООП - інкапсуляція, старт ООП, 4(перегрузка), 5(наследование, полиморфизм)
//сделать клас который будет с ним работать
//
```

```
namespace Lab3OP;
```

```
class Program
{
```

```
    static void Main(string[] args)
    {
        Worker worker = new Worker();
        Train[] trains = worker.Generator();
        Train[] DiapasonTrains=worker.GetTrainsDiapason(trains, worker.GetDiapason());
        Console.Clear();
        Console.WriteLine("-----\nAll trains");
        worker.PrintArray(trains);
        Console.WriteLine("-----\nTrains in Diapason");
```

```
worker.PrintArray(DiapasonTrains);
```

```
}
```

```
}
```

Train.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab3OP
{ //Класс потяг
    public class Train
    {

        public int Number { get; private set; }
        public string Destination { get; private set; }
        public TimeOnly TimeStart { get; private set; }

        //Конструктор объекта класу
        public Train(int number, string destination, TimeOnly timeStart)
        {
            Number = number;
            Destination = destination;
            TimeStart = timeStart;
        }

        public override string ToString()
        {
            return $"Train number - {Number} to {Destination} at {TimeStart}";
        }

    }
}
```

Worker.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab3OP
{
    public class Worker
    {

        //Створення потягів та запис їх у масив
        public Train[] Generator()
        {

            Console.WriteLine("Please enter amount of trains do you want.");
            int amount = Convert.ToInt32(Console.ReadLine());
            Train[] trains = new Train[amount];
            Random random = new Random();
            for (int i = 0; i < amount; i++)
            {
                int number = random.Next(1111, 10000);
                Console.WriteLine("Enter destination of your train");
                string destination = Console.ReadLine();
                Console.WriteLine("Enter time of start train run in format HH:MM");
                TimeOnly start = TimeOnly.Parse(Console.ReadLine());
                trains[i] = new Train(number, destination, start);

            }
            return trains;
        }

        //Отримання введення діапазону годину від користувача
        public (TimeOnly start, TimeOnly end) GetDiapason()
        {
            Console.WriteLine("Enter starttime of your diapason in HH:MM format");
            TimeOnly start = TimeOnly.Parse(Console.ReadLine());
            Console.WriteLine("Enter endtime of your diapason in HH:MM format");
            TimeOnly end = TimeOnly.Parse(Console.ReadLine());
            return (start, end);
        }

        //Виведення масиву
        public void PrintArray(Train[] trains)
        {
            foreach (Train train in trains)
            {
                Console.WriteLine(train.ToString());
            }
        }

        //Перевірка на діапазон годин
        public bool CheckDiapason(Train train, (TimeOnly start, TimeOnly end) Diapason)
        {
            //22 - 7 6 6<22 6

            if (Diapason.start < Diapason.end)
            {

                return train.TimeStart >= Diapason.start && train.TimeStart <= Diapason.end;

            }
        }
    }
}

```

```

else // 20<=19 && 20>=23 6 19 - 7 6>=19 &&
{

    if (train.TimeStart >= Diapason.start) return true;
    else if (train.TimeStart <= Diapason.end) return true;
    else return false;
}

}
//Перевірка потягу щодо належності до часового проміжку та запис при належності у масив
public Train[] GetTrainsDiapason(Train[] trains, (TimeOnly start, TimeOnly end) Diapason)
{

    List<Train> trainsDiapason = new List<Train>();
    foreach (Train train in trains)
    {
        if (CheckDiapason(train, Diapason))
        {
            trainsDiapason.Add(train);
        }
    }
    return trainsDiapason.ToArray();
}
}
}

```

Висновок

Отже, я вивчив механізми створення та використання класів та об'єктів. Мною було розроблено клас потяг та його характеристики, реалізовано утворення масиву об'єктів даного класу, отримання користувацького вводу для утворення цих потягів та діапазону часу для виділення тих, які входять у нього. Після утворення масиву потягів була реалізована їх обробка для перевірки належності до часового періоду відправлення потягів заданого користувачем. Після написання коду на мові C# програма була протестована, були виправлені помилки та першочергові недоліки алгоритму. У висновку завдання є виконаним, а набуті теоретичні знання були закріплені за рахунок виконання поставленого завдання.