

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА ИИТ

Лабораторная работа №4

По дисциплине: «Современные платформы программирования»

Выполнил:

Студент 3 курса

группы ПО-8:

Печко В.И.

Проверил:

Крощенко А.А.

Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

Вариант 18

Задание 1.

9) Создать класс Mobile с внутренним классом, с помощью объектов которого можно хранить информацию о моделях телефонов и их свойствах..

Код программы:

Mobile.cs:

```
using System;
using System.Collections.Generic;

namespace Task1
{
    public class Mobile
    {
        private class Model
        {
            public string Name { get; }
            public string Properties { get; }

            public Model(string name, string properties)
            {
                Name = name;
                Properties = properties;
            }
        }

        private Dictionary<string, List<Model>> models;

        public Mobile()
        {
            models = new Dictionary<string, List<Model>>();
        }

        public void AddModel(string name, string properties)
        {
            if (!models.ContainsKey(name))
            {
                models[name] = new List<Model>();
            }

            models[name].Add(new Model(name, properties));
        }

        public List<string> FindModels(string name)
        {
            if (models.ContainsKey(name))
```

```

        {
            List<string> foundModels = new List<string>();
            foreach (var model in models[name])
            {
                foundModels.Add(model.Properties);
            }
            return foundModels;
        }
        else
        {
            return new List<string>();
        }
    }

    public void PrintAllModels()
    {
        foreach (var entry in models)
        {
            Console.WriteLine($"Model: {entry.Key}");
            foreach (var model in entry.Value)
            {
                Console.WriteLine($"- {model.Properties}");
            }
            Console.WriteLine();
        }
    }
}

Program.cs:
using Task1;

Mobile myMobile = new Mobile();
myMobile.AddModel("iPhone 13", "Properties: A15 Bionic, 6.1\" Super Retina XDR display");
myMobile.AddModel("Samsung Galaxy S21", "Properties: Exynos 2100, 6.2\" Dynamic AMOLED 2X");
List<string> iPhoneModels = myMobile.FindModels("iPhone 13");
foreach (var model in iPhoneModels)
{
    Console.WriteLine("- " + model);
}
Console.WriteLine("All Models:");
myMobile.PrintAllModels();

```

Результат программы:

```

- Properties: A15 Bionic, 6.1" Super Retina XDR display
All Models:
Model: iPhone 13
- Properties: A15 Bionic, 6.1" Super Retina XDR display
Model: Samsung Galaxy S21
- Properties: Exynos 2100, 6.2" Dynamic AMOLED 2X

```

Задание 2.

6) Создать класс Страница, используя класс Слово.

Код программы:

Page.cs:

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

namespace Task_2
{
    internal class Page
    {
        private List<Word> words;

        public Page()
        {
            this.words = new List<Word>();
        }

        public int WordCount()
        {
            return words.Count;
        }

        public List<Word> GetWords()
        {
            return words;
        }

        public void AddWord(Word word)
        {
            words.Add(word);
        }
    }
}
```

Program.cs:

```
using System;
using System.Threading.Tasks;

class Program
{
    static void Main(string[] args)
    {
        Task_2.Page page = new Task_2.Page();

        Console.WriteLine("Enter words for a page (separated by space), or type 'done' to finish:");
```

```

    bool addingWords = true;
    while (addingWords)
    {
        string input = Console.ReadLine();
        if (input.ToLower() == "done")
        {
            addingWords = false;
        }
        else
        {
            string[] wordStrings = input.Split(' ');
            foreach (string wordString in wordStrings)
            {
                page.AddWord(new Task_2.Word(wordString));
            }
        }
    }

    Console.WriteLine("Number of words on the page: " + page.WordCount());
}
}

```

Word.cs:

```

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

```

```

namespace Task_2
{
    internal class Word
    {
        private string text;

        public Word(string text)
        {
            this.text = text;
        }

        public string GetText()
        {
            return text;
        }
    }
}

```

Результат программы:

```
Enter words for a page (separated by space), or type 'done' to finish:
The quick brown fox jumps over the lazy dog
done
Number of words on the page: 9
```

Задание 3.

6) Система Автобаза. Диспетчер распределяет заявки на Рейсы между Водителями и назначает для этого Автомобиль. Водитель может сделать заявку на ремонт. Диспетчер может отстранить Водителя от работы. Водитель делает отметку о выполнении Рейса и состоянии Автомобиля.

Код программы:

Car.cs:

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

namespace Task_3
{
    internal class Car
    {
        public int Id { get; }
        public string Model { get; }

        public Car(int id, string model)
        {
            Id = id;
            Model = model;
        }
    }
}
```

Dispatcher.cs:

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

namespace Task_3
{
    internal class Dispatcher
    {
        public int Id { get; }
        public string Name { get; }
        public List<Driver> Drivers { get; }
        public List<Car> Cars { get; }

        public Dispatcher(int id, string name)
```

```

    {
        Id = id;
        Name = name;
        Drivers = new List<Driver>();
        Cars = new List<Car>();
    }

    public void AssignCarToDriver(Driver driver, Car car)
    {
        driver.Car = car;
        Console.WriteLine($"Car {car.Model} assigned to driver {driver.Name}.");
    }

    public void SuspendDriver(Driver driver)
    {
        driver.IsSuspended = true;
        Console.WriteLine($"Driver {driver.Name} is suspended from work.");
    }

    public void RegisterTrip(Trip trip)
    {
        trip.IsCompleted = true;
        Console.WriteLine($"Trip {trip.Id} has been completed by driver {trip.Driver.Name}.");
    }
}

```

Driver.cs:

```

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

namespace Task_3
{
    internal class Driver
    {
        public int Id { get; }
        public string Name { get; }
        public Car Car { get; set; }
        public bool IsSuspended { get; set; }

        public Driver(int id, string name)
        {
            Id = id;
            Name = name;
            IsSuspended = false;
        }

        public void RequestRepair()
        {
            Console.WriteLine($"Driver {Name} has requested a repair for car {Car.Model}.");
        }
    }
}

```

```

    }

    public void CompleteTrip(Trip trip)
    {
        trip.IsCompleted = true;
        Console.WriteLine($"Driver {Name} has completed trip {trip.Id}.");
    }
}

```

Trip.cs:

```

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

namespace Task_3
{
    internal class Trip
    {
        public int Id { get; }
        public Driver Driver { get; }
        public bool IsCompleted { get; set; }

        public Trip(int id, Driver driver)
        {
            Id = id;
            Driver = driver;
            IsCompleted = false;
        }
    }
}

```

Program.cs:

```

using System;
using Task_3;

namespace Task_3
{
    class Program
    {
        static void Main(string[] args)
        {
            Dispatcher dispatcher = new Dispatcher(1, "Dispatcher1");
            Driver driver1 = new Driver(1, "Driver1");
            Car car1 = new Car(1, "Car1");
            Trip trip1 = new Trip(1, driver1);

            // Диспетчер назначает автомобиль водителю
            dispatcher.AssignCarToDriver(driver1, car1);
        }
    }
}

```



```
// Водитель выполняет рейс
driver1.CompleteTrip(trip1);

// Диспетчер регистрирует выполненный рейс
dispatcher.RegisterTrip(trip1);
}
}
}
```

Результат программы:

```
Car Car1 assigned to driver Driver1.
Driver Driver1 has completed trip 1.
Trip 1 has been completed by driver Driver1.
```

Вывод: приобрести практические навыки в области объектно-ориентированного проектирования.