

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра технічної кібернетики

### **Лабораторна робота № 3**

по дисципліні «Об'єктно-орієнтоване програмування»

Тема: Робота з конструкторами класів «Транспорт2»

Виконав:

студент групи ІТ-62

ПІБ Любченко П.І.

Дата здачі \_\_\_\_\_

Захищено з балом \_\_\_\_\_

Перевірено:

ст.вик. кафедри АУТС

Хмелюк Марина Сергіївна

Київ 2018

Завдання:

Транспорт2 (водитель, перевозки, маршрут, тип транспорта)

Согласно теме:

- в одном из классов реализовать конструктор без параметров, деструктор;
- в одном из классов реализовать конструктор с параметрами, деструктор;
- одним из классов реализовать конструктор, в котором создаются другие объекты этого же класса.
- реализовать перегруженный конструктор;
- написать отчет (текст программы, результат работы, выводы).

Код конструкторів класів:

```
public Driver(string firstname, string secondname)
{
    FirstName = firstname;
    SecondName = secondname;
    Drivers.Add(this);
}
public Driver()
{
    FirstName = "";
    SecondName = "";
    Drivers.Add(this);
}
public Driver(string[] firstnames, string secondname)
{
    if (firstnames.Length < 1)
        throw new ArgumentException("Empty first name");
    FirstName = firstnames[0];
    SecondName = secondname;
    Drivers.Add(this);
    for (int i = 1; i < firstnames.Length - 1; i++)
        new Driver(firstnames[i], secondname);
}
public Driver(string firstname, string[] secondnames)
{
    if (secondnames.Length < 1)
        throw new ArgumentException("Empty first name");
    FirstName = firstname;
    SecondName = secondnames[0];
    Drivers.Add(this);
    for (int i = 1; i < secondnames.Length - 1; i++)
        new Driver(firstname, secondnames[i]);
}

public Route(TransportType transporttype)
{
    TransportType = transporttype;
    Number = Routes.Count >= 1 ? Routes.Count + 1 : 1;

    Routes.Add(this);
}
public Route(string cargo)
{
    var tt = new TransportType(cargo, cargo);
    new Route(tt);
}
```

```

public RouteTransportationAgreement(Route route, Transportation transportation)
{
    RouteId = route.GetId();
    TransportationId = transportation.GetId();
    Agreements.Add(this);
}

public TransportType(string name, string cargo)
{
    Name = name;
    Cargo = cargo;
    bool isExist = false;
    foreach (var tt in TransportTypes)
        if (tt.Name == name && tt.Cargo == cargo)
        {
            isExist = true;
            foreach (Transportation tr in Transportation.Transportations)
                if (tr.TransportType == this)
                    tr.ChangeTransportType(tt);
            break;
        }
    if (!isExist)
        TransportTypes.Add(this);
}

public TransportType()
{
    Name = String.Empty;
    Cargo = String.Empty;

    TransportTypes.Add(this);
}

```

Код деструкторів:

```

~Driver()
{
    License.Delete(this);
    Route = null;
    FirstName = null;
    SecondName = null;
}

~Route()
{
    RouteTransportationAgreement.Delete(this);
    TransportType = null;
}

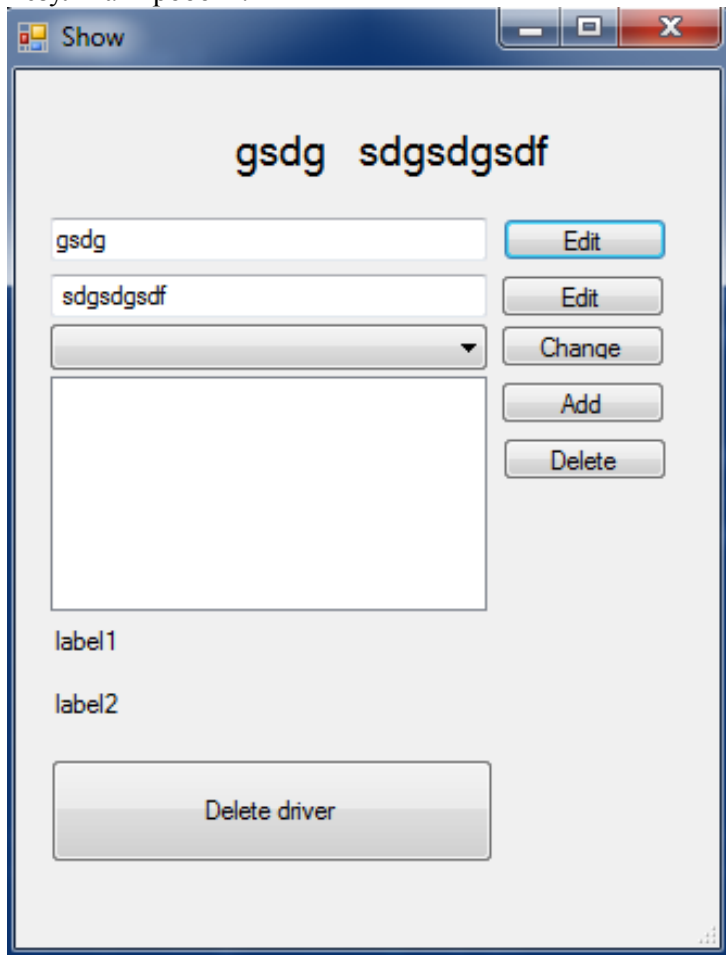
~RouteTransportationAgreement()
{
    RouteId = Guid.Empty;
    TransportationId = Guid.Empty;
}

~Transportation()
{
    TransportType = null;
    RouteTransportationAgreement.Delete(this);
}

```

```
~TransportType()  
{  
    License.Delete(this);  
    Name = null;  
    Cargo = null;  
}
```

Результати роботи:



Висновок: я покращив вміння створювати конструктори та деструктори, дізнався про деякі сфери їх використання.

