

Министерство образования Республики Беларусь  
Учреждение образования «Брестский государственный технический университет»  
Кафедра ИИТ

**Лабораторная работа №6**  
по ООПиП  
**«НАСЛЕДОВАНИЕ И ВИРТУАЛЬНЫЕ ФУНКЦИИ»**

Выполнила: студентка 2-го курса  
группы АС-53 Замулко Д.И.  
Проверила: Давидюк Ю.И.

Брест, 2020

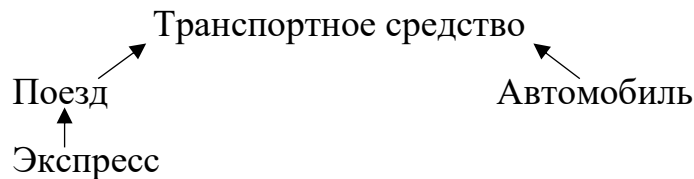
## Лабораторная работа №6 «НАСЛЕДОВАНИЕ И ВИРТУАЛЬНЫЕ ФУНКЦИИ»

**Цель:** получить практические навыки создания иерархии классов и использования статических компонентов класса.

### Вариант 11

Написать программу, в которой создается иерархия классов. Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

11) автомобиль, поезд, транспортное средство, экспресс.



```
#include <iostream>
#include <string>
using namespace std;

class TranspVehicle {
public:
    static TranspVehicle* start;
    TranspVehicle* next = NULL;

    static void ShowList() {
        TranspVehicle* p = start;
        while (p) {
            p->show();
            p = p->next;
        }
    }

    TranspVehicle() {
        cout << "--Default constructor--" << endl;
    }

    TranspVehicle(string newName, double newExperience, double newSpeed) {
        name = newName;
        experience = newExperience;
        speed = newSpeed;
    }

    virtual ~TranspVehicle() {
        cout << "--Default destructor--" << endl;
    }

    virtual void show() = 0;
    virtual void input() = 0;
    virtual void addToList() = 0;
protected:
    string name;
    double experience;
```

```

        double speed;
    };

class Train :public TranspVehicle {
public:
    Train() : TranspVehicle() {};
    Train(string nameOfRailwayN, int numberN, double speedN,
        string nameN, double experienceN) {
        number = numberN;
        speed = speedN;
        name = nameN;
        experience = experienceN;
        nameOfRailway = nameOfRailwayN;
    }

    void show() {
        cout << "---Train---" << endl;
        cout << " The name of the railway: " << nameOfRailway << endl;
        cout << " Number : " << number << endl;
        cout << " Max speed: " << speed << endl;
        cout << " Driver's name: " << name << endl;
        cout << " Experience of work: " << experience << endl;
        cout << endl;
    }

    void input() {
        cout << "---Train---" << endl;
        cout << " Enter the name of the railway: "; cin >> nameOfRailway;
        cout << " Enter number: "; cin >> number;
        cout << " Enter max speed: "; cin >> speed;
        cout << " Enter driver's name: "; cin >> name;
        cout << " Enter experience of work: "; cin >> experience;
        cout << endl;
    }

    void addToList() {
        TranspVehicle* p = start;
        while (p->next) {
            p = p->next;
        }
        p->next = this;
    }
protected:
    int number;
    string nameOfRailway;
};

class Express :public Train {
public:
    Express() : Train() {};
    Express(string nameOfRailwayN, int numberN, double speedN,
        string nameOfOrganizationN, string nameN, double experienceN) {
        nameOfRailway = nameOfRailwayN;
        number = numberN;
        speed = speedN;
        nameOfOrganization = nameOfOrganizationN;
        name = nameN;
        experience = experienceN;
    }
}

```

```

void show() {
    cout << "---Express---" << endl;
    cout << " The name of the railway:" << nameOfRailway << endl;
    cout << " Number: " << number << endl;
    cout << " Max speed: " << speed << endl;
    cout << " Name of organization: " << nameOfOrganization << endl;
    cout << " Driver's name: " << name << endl;
    cout << " Experience of work: " << experience << endl;
    cout << endl;
}

void input() {
    cout << "---Express---" << endl;
    cout << " Enter the name of the railway: "; cin >> nameOfRailway;
    cout << " Enter number: "; cin >> number;
    cout << " Enter max speed: "; cin >> speed;
    cout << " Enter name of organization: "; cin >> nameOfOrganization;
    cout << " Enter driver's name: "; cin >> name;
    cout << " Enter experience of work: "; cin >> experience;
    cout << endl;
}

void addToList() {
    TranspVehicle* p = start;
    while (p->next) {
        p = p->next;
    }
    p->next = this;
}

private:
    string nameOfOrganization;
};

class Car :public TranspVehicle {
public:
    Car() : TranspVehicle() {};
    Car(string typeN, string carBrandN, double speedN, string nameN,
        double experienceN) {
        type = typeN;
        carBrand = carBrandN;
        speed = speedN;
        name = nameN;
        experience = experienceN;
    }

    void show() {
        cout << "---Car---" << endl;
        cout << " Type of car: " << type << endl;
        cout << " Car brand: " << carBrand << endl;
        cout << " Max speed: " << speed << endl;
        cout << " Driver's name: " << name << endl;
        cout << " Experience of work: " << experience << endl;
        cout << endl;
    }

    void input() {
        cout << "---Car---" << endl;
        cout << " Enter type of car: "; cin >> type;
        cout << " Enter car brand: "; cin >> carBrand;
        cout << " Max speed: " << speed << endl;
        cout << " Enter driver's name: "; cin >> name;
    }
}

```

```

        cout << " Enter experience of work: "; cin >> experience;
        cout << endl;
    }

    void addToList() {
        TranspVehicle* p = start;
        while (p->next) {
            p = p->next;
        }
        p->next = this;
    }
protected:
    string type;
    string carBrand;
};
TranspVehicle* TranspVehicle::start = NULL;
int main() {
    Train* train;
    Car* car;
    Express* express;
    train = new Train();
    car = new Car();
    express = new Express();
    train->input();
    car->input();
    express->input();
    TranspVehicle::start = train;
    car->addToList();
    express->addToList();
    TranspVehicle::ShowList();
}

```

The image shows two side-by-side windows of the Microsoft Visual Studio debug console. The left window, titled 'Консоль отладки Microsoft Visual Studio', displays the program's execution flow with prompts and user input. It shows the creation of Train, Car, and Express objects and their addition to a linked list. The right window, also titled 'Консоль отладки Microsoft Visual Studio', displays the output of the ShowList() method, showing the details of each object in the list: Train (BelRailway, speed 120, Driver1, 10 years experience), Car (automobile, brand BMW, speed 160, Driver2, 5 years experience), and Express (RussRailway, speed 140, Rus1 organization, Driver3, 20 years experience).

```

--Default constructor--
--Default constructor--
--Default constructor--
---Train---
Enter the name of the railway: BelRailway
Enter number: 1
Enter max speed: 120
Enter driver's name: Driver1
Enter experience of work: 10

---Car---
Enter type of car: automobile
Enter car brand: bmw
Enter max speed: 160
Enter driver's name: Driver2
Enter experience of work: 5

---Express---
Enter the name of the railway: RussRailway
Enter number: 2
Enter max speed: 140
Enter name of organization: Rus1
Enter driver's name: Driver3
Enter experience of work: 20

---Train---
The name of the railway: BelRailway
Number: 1
Max speed: 120
Driver's name: Driver1
Experience of work: 10

---Car---
Type of car: automobile
Car brand: bmw
Max speed: 160
Driver's name: Driver2
Experience of work: 5

---Express---
The name of the railway: RussRailway
Number: 2
Max speed: 140
Name of organization: Rus1
Driver's name: Driver3
Experience of work: 20

```

**Вывод:** в ходе лабораторной работы получила практические навыки создания иерархии классов и использования статических компонентов класса.