

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.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<iostream>
#include<Windows.h>
#include<string>

using namespace std;

class TranspVehicle // базовый класс
{
public:
    static TranspVehicle* beg; //указатель на начало списка
    TranspVehicle* next = NULL;
    static void ShowList() //список
    {
        TranspVehicle* p = beg;
        while (p)
        {
            p->show();
            p = p->next;
        }
    }
    TranspVehicle() //без парам
    {
        name = new char[81];
    }
    TranspVehicle(const char* NAME, int EXPERIENCE) //с парам
    {
        // выделение памяти для name. размер выделяемой памяти = длина строки NAME
        name = new char[strlen(NAME) + 1];
        strcpy(name, NAME);
        experience = EXPERIENCE;
    }
    virtual ~TranspVehicle() // виртуальный деструктор
    {
        delete[] name;
    }
    virtual void show() = 0; //Чистая виртуальная функция
    virtual void input() = 0;
```

```

protected:
    char* name;
    int experience;
};

TranspVehicle* TranspVehicle::beg = NULL; //Инициализация статической компоненты

class car :public TranspVehicle // производный класс
{
public:
    car() : TranspVehicle() {} //без парам
    car(const char* STAMP, int EXPERIENCE, bool AddToList = false) :TranspVehicle(STAMP,
EXPERIENCE) //с парам
    {
        if (AddToList)
        {
            TranspVehicle* p = beg;
            while (p->next)
            {
                p = p->next;
            }
            p->next = this;
        }
    }
    void show()
    {
        cout << "¥n Класс: Автомобиль";
        cout << "¥n Имя водителя: " << name;
        cout << "¥n Водительский стаж:" << experience;
        cout << "¥n";
    }
    void input()
    {
        cout << "¥n Имя водителя АВТОМОБИЛЯ: ";
        cin >> name;
        cout << "¥n Водительский стаж:";
        cin >> experience;
        cout << "¥n";
        cout << "-----¥n";
    }
};

class train :public TranspVehicle // производный класс
{
public:
    train() : TranspVehicle() {}

    train(const char* NAME, int EXPERIENCE, bool AddToList = false) :TranspVehicle(NAME,
EXPERIENCE)
    {
        if (AddToList)
        {
            TranspVehicle* p = beg;
            while (p->next)
            {
                p = p->next;
            }
            p->next = this;
        }
    }
}

```

```

void show()
{
    cout << "¥nКласс: Поезд";
    cout << "¥n Имя водителя: " << name;
    cout << "¥n Водительский стаж:" << experience;
    cout << "¥n";
}
void input()
{
    cout << "¥n Имя водителя ПОЕЗДА: ";
    cin >> name;
    cout << "¥n Водительский стаж:";
    cin >> experience;
    cout << "¥n";
    cout << "-----¥n";
}

};

class express :public train // производный класс
{
public:
    express() : train() {}

    express(const char* NAME, int EXPERIENCE, bool AddToList = false) :train(NAME,
EXPERIENCE)
    {
        if (AddToList)
        {
            TranspVehicle* p = beg;
            while (p->next)
            {
                p = p->next;
            }
            p->next = this;
        }
    }
    void show()
    {
        cout << "¥n Класс: Экспресс";
        cout << "¥n Имя водителя: " << name;
        cout << "¥n Водительский стаж:" << experience;
        cout << "¥n";
    }
    void input()
    {
        cout << "¥n Имя водителя ЭКСПРЕСС: ";
        cin >> name;
        cout << "¥n Водительский стаж:";
        cin >> experience;
        cout << "¥n";
    }
}

};

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    car* x1;
    train* x2;
    express* x3;

```

```

x1 = new car;
x2 = new train;
x3 = new express;
x1->input();
x2->input();
x3->input();
cout << "_____ \n";
TranspVehicle::beg = x1;
x1->next = x2;
x2->next = x3;
express* x4 = new express("Игорь", 5, true); // Создание объекта класса
TranspVehicle::ShowList();
return 0;
}

```

```

Консоль отладки Microsoft Visual Studio

Имя водителя АВТОМОБИЛЯ: Владимир
Водительский стаж:4
-----

Имя водителя ПОЕЗДА: Николай
Водительский стаж:5
-----

Имя водителя ЭКСПРЕСС: Дмитрий
Водительский стаж:6

Класс: Автомобиль
Имя водителя: Владимир
Водительский стаж:4

Класс: Поезд
Имя водителя: Николай
Водительский стаж:5

Класс: Экспресс
Имя водителя: Дмитрий
Водительский стаж:6

Класс: Экспресс
Имя водителя: Игорь
Водительский стаж:5

C:\Users\Dasha Zamulko\source\repos\Project25\Debug\Project25.exe (процесс

```

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