

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №2 по курсу**  
**«Проектирование программ в интеллектуальных системах»**

**Выполнили студенты группы**

**921701:**

**Проверил:**

**Соловьев А.**

**Садовский М.Е**

**МИНСК**

**2020**

**Цель работы:** Изучить основы наследование на языке программирования C++.

**Задание:**

**Длинное целое со знаком.**

Каждому студенту требуется выбрать предметную область и согласовать её с преподавателем. В рамках выбранной предметной области следует построить иерархию классов с применением следующих концепций ООП:

- закрытое
- открытое
- защищённое наследования
- виртуальные методы
- разрешение имён с помощью using-директивы
- множественное наследование
- виртуальное наследование.

## Ход и результаты работы:

В программе реализована иерархия классов. Вариант – вооружение. Самым старшим классом является класс Armament.

```
#pragma once
#include <iostream>
#include <string>

using namespace std;

class Armament {
protected:
    string manufacturer_country;
    string name_of_armament;
    int cost;
public:
    void Get_manufacturer_country(string country);
    void Get_name_of_armament(string name);
    void Get_cost(int price);
    virtual void View() = 0;
};

class Ship {
protected:
    int displacement;
public:
    void Get_displacement(int volume);
};
```

В нём находятся несколько protected полей и public методов. Также он включает в себя виртуальную функцию View(), которая задействуется во всех ниже стоящих классах.

В программе реализовано множественное наследование.

```
class Battle_ship :public Technics,public Ship {
public:
    void View();
};

class Battle_plan :public Technics, public Plan {
public:
    void View();
};

class Battle_car :public Technics, public Car {
public:
    void View();
};
```

В этом примере видно три класса получившиеся смешанным наследованием. В данном случае они даже не имеют своих полей и методов, кроме функции просмотра. Здесь классы наследуются через public, это значит, что все protected поля наследуются в protected, public методы также сохраняют свой уровень доступа. Privet поля не наследуются!

В программе реализовано пространство имён Explosive\_device.

```
namespace Explosive_device {
    class Grenade :virtual public Armament {
protected:
    int affected_area;
public:
    void View();
    void Get_affected_area(int radius);
    };
}
```

Таким образом мы изолируем класс Grenade. В дальнейшем, чтобы получить к нему доступ требуется сделать объявление типа:

```
120     using namespace Explosive_device;
121     Armament* pointer;
122     Assault_rifle ak74;
123     Rifle m24;
124     Grenade f1;
125     Battle_ship bismarck;
126     Battle_plan i16;
127     Battle_car lb23;
128     Grenade_launcher rpg7;
```

Через using-директиву можно открыть доступ к полю имени в целом блоке.

Однако есть способ открыть доступ локально:

```
, Explosive_device::Grenade& f1,
```

Пример виртуального наследования

```
class Rifle :virtual public Firearm {
protected:
    int range_of_defeat;
public:
    void Get_range_of_defeat(int range);
    void View();
};

namespace Explosive_device {
class Grenade :virtual public Armament {
protected:
    int affected_area;
public:
    void View();
    void Get_affected_area(int radius);
};
}
```

Он нужен для того, чтобы в коде программы не образовалось неопределённости. Если класс наследуется от классов, у которых есть общие базовые классы.

## Вывод

Наследование (англ. inheritance) — концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения.

Наследование является механизмом повторного использования кода (англ. code reuse) и способствует независимому расширению программного обеспечения через открытые классы (англ. public classes) и интерфейсы (англ. interfaces). Установка отношения наследования между классами порождает иерархию классов (англ. class hierarchy).