|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 8 |

**Название:**

Наследование.

**Дисциплина:** Объектно-ориентированное программирование

Вариант 16.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-21Б |  |  | Т.Е.Старжевский |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | О.А.Веселовская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

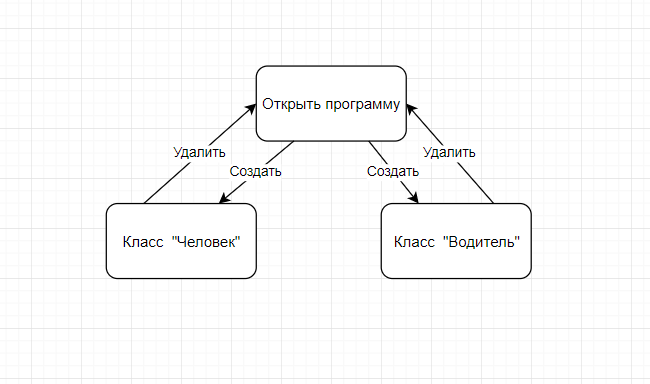
**Лабораторная работа 8. Наследование**

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Протестировать все методы каждого класса. Все поля классов должны быть скрытыми (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

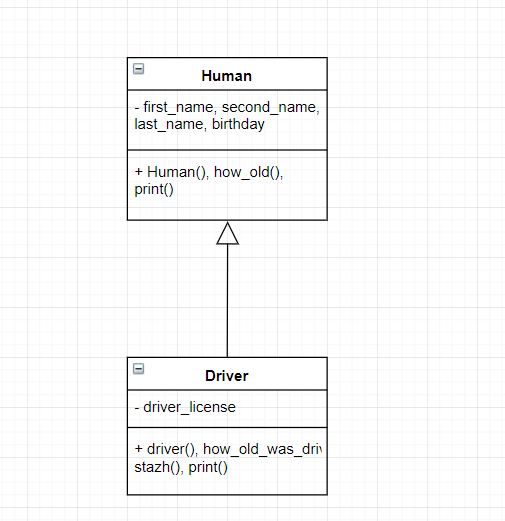
Объект – человек. Поля: ФИО, год рождения. Методы: процедура инициализации объекта, процедура вывода информации об объекте на экран, функция вычисления возраста человека.

Объект – автомобилист. Поля: ФИО, год рождения, год получения водительских прав. Методы: процедура инициализации объекта, процедура вывода информации об объекте на экран, функция вычисления водительского стажа и функция вычисления возраста на момент получения прав.

**Объектная декомпозиция:**

****

**Диаграмма классов предметной области:**

****

**Код программы:**

#pragma once

#include <iostream>

using namespace std;

const int today\_year = 2022;

struct date

{

int day, month, year;

};

class human

{

protected:

string first\_name, second\_name, last\_name;

date birthday;

public:

human(string ifirst\_name, string isecond\_name, string ilast\_name, date ibirthday) : first\_name(ifirst\_name), second\_name(isecond\_name),

last\_name(ilast\_name)

{

birthday.day = ibirthday.day;

birthday.month = ibirthday.month;

birthday.year = ibirthday.year;

};

human() :human("none", "none", "none", { 0, 0, 0 }) {};

int how\_old()

{

return today\_year - birthday.year;

};

void print()

{

cout << "Name: " << first\_name << ' ' << second\_name << ' ' << last\_name << "\tBirthday: "

<< birthday.day << '/' << birthday.month << '/' << birthday.year << "\t" << how\_old() << " Years old";

}

};

class driver final : public human

{

protected:

date driver\_license;

public:

driver(string ifirst\_name, string isecond\_name, string ilast\_name, date ibirthday, date idriver\_license) :

human(ifirst\_name, isecond\_name, ilast\_name, ibirthday)

{

driver\_license = idriver\_license;

};

driver() : driver("none", "none", "none", { 0, 0, 0 }, { 0, 0, 0 }) {};

int stazh()

{

return today\_year - driver\_license.year;

};

int how\_old\_was\_driver()

{

return driver\_license.year - birthday.year;

};

void print()

{

cout << "Name: " << first\_name << ' ' << second\_name << ' ' << last\_name << "\tBirthday: "

<< birthday.day << '/' << birthday.month << '/' << birthday.year << "\t" << how\_old() << " Years old\n"

<< "Driver license from: " << driver\_license.day << '/' << driver\_license.month << '/' << driver\_license.year << "\tWere received in "

<< how\_old\_was\_driver() << " years" << "\nStazh right now: " << stazh() << " years";

};

};

#include <iostream>

#include "class.h"

using namespace std;

int main()

{

human Timofey("Starzhevsky", "Timofey", "Evgenyevich", {21, 9, 2003 } );

driver Evgeny("Starzhevsky", "Evgeny", "Viktorovich", { 15, 4, 1976 }, { 1,1,2010 });

Timofey.print();

cout << endl;

Evgeny.print();

return 0;

};

**Тестирование:**

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Полученный результат |
| Объекты Тимофей, Евгений | Вывод информации об объектах |  |

При заданных значениях программа работает корректно.

**Вывод:** Научился работать с наследуемыми объектами на языке C++.