

Цель работы: целью данной работы была разработка программы, демонстрирующей использование механизмов наследования для создания иерархии классов, описывающих объекты "человек" и "студент".

Задание: Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Составить программный код и написать тестирующую программу.

Все поля классов должны быть частными (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: конструктор, процедура вывода содержимого полей на экран и функция, определяющая, какой был (или будет) возраст у человека в заданном году.

Объект — студент. Поля: ФИО, год рождения, год поступления в вуз, название вуза. Методы: конструктор, процедура вывода информации о студенте на экран и функция, вычисляющая возраст студента на момент поступления в вуз.

Выполнение: Диаграммы классов Human и Student изображены на рисунках 1 и 2 соответственно

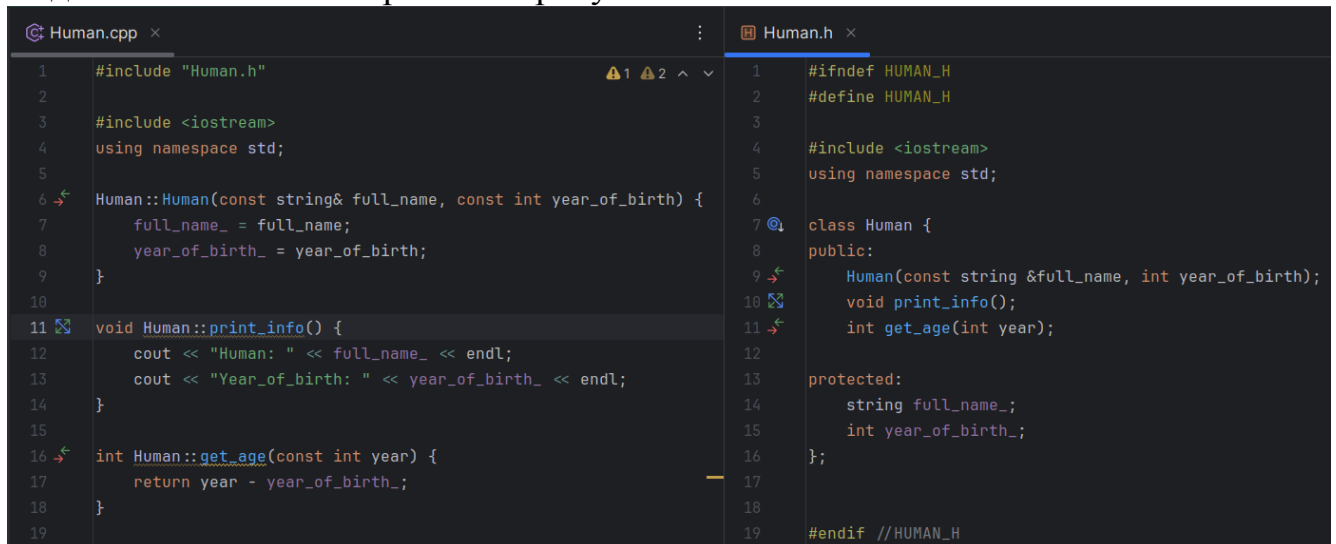
Human
full_name_, year_of_birth_
print_info(), get_age(int year)

Рисунок 1 — диаграмма класса Human

Student
full_name_, year_of_birth_, admission_to_university_year_, university_name_
print_info(), get_age(int year), get_admission_age()

Рисунок 2 — диаграмма класса Student

Код класса Human изображен на рисунке 3

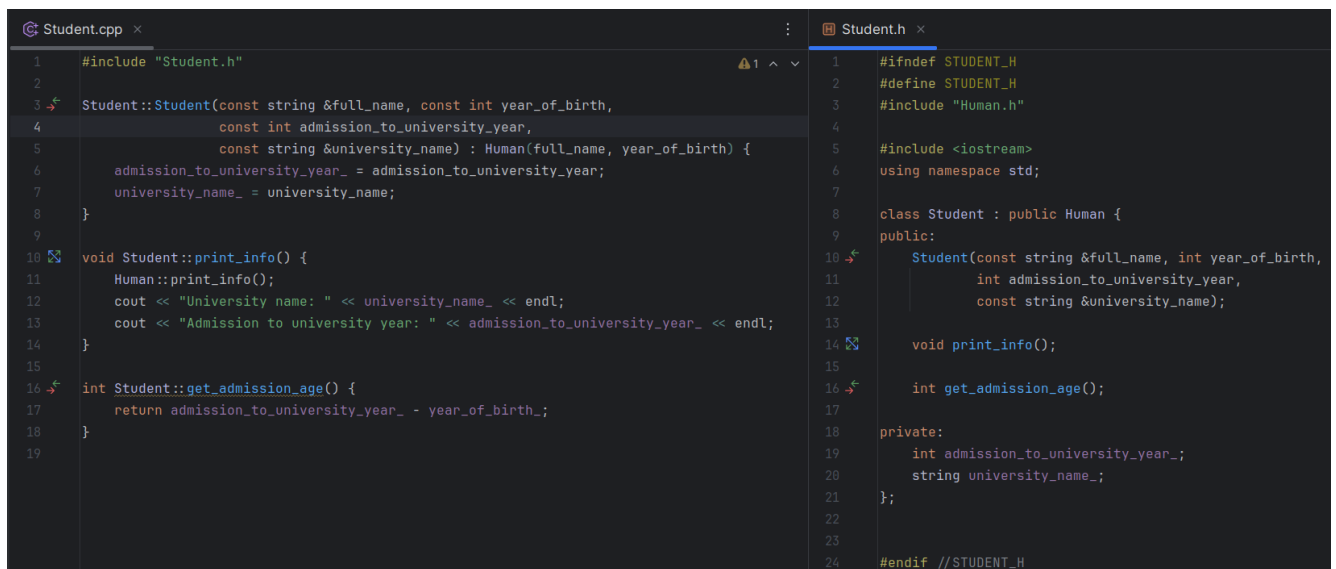


```
Human.cpp
1 #include "Human.h"
2
3 #include <iostream>
4 using namespace std;
5
6 Human::Human(const string& full_name, const int year_of_birth) {
7     full_name_ = full_name;
8     year_of_birth_ = year_of_birth;
9 }
10
11 void Human::print_info() {
12     cout << "Human: " << full_name_ << endl;
13     cout << "Year_of_birth: " << year_of_birth_ << endl;
14 }
15
16 int Human::get_age(const int year) {
17     return year - year_of_birth_;
18 }
19

Human.h
1 #ifndef HUMAN_H
2 #define HUMAN_H
3
4 #include <iostream>
5 using namespace std;
6
7 class Human {
8 public:
9     Human(const string &full_name, int year_of_birth);
10    void print_info();
11    int get_age(int year);
12
13 protected:
14     string full_name_;
15     int year_of_birth_;
16 };
17
18
19 #endif //HUMAN_H
```

Рисунок 3 — код класса Human

Код класса Student изображен на рисунке 4



```
Student.cpp
1 #include "Student.h"
2
3 Student::Student(const string &full_name, const int year_of_birth,
4                 const int admission_to_university_year,
5                 const string &university_name) : Human(full_name, year_of_birth) {
6     admission_to_university_year_ = admission_to_university_year;
7     university_name_ = university_name;
8 }
9
10 void Student::print_info() {
11     Human::print_info();
12     cout << "University name: " << university_name_ << endl;
13     cout << "Admission to university year: " << admission_to_university_year_ << endl;
14 }
15
16 int Student::get_admission_age() {
17     return admission_to_university_year_ - year_of_birth_;
18 }
19

Student.h
1 #ifndef STUDENT_H
2 #define STUDENT_H
3 #include "Human.h"
4
5 #include <iostream>
6 using namespace std;
7
8 class Student : public Human {
9 public:
10    Student(const string &full_name, int year_of_birth,
11           int admission_to_university_year,
12           const string &university_name);
13
14    void print_info();
15
16    int get_admission_age();
17
18 private:
19    int admission_to_university_year_;
20    string university_name_;
21 };
22
23
24 #endif //STUDENT_H
```

Рисунок 4 — код класса Student

Текст основной (тестирующей) программы изображен на рисунке 5

```
#include <iostream>

#include "Human.h"
#include "Student.h"
using namespace std;

int main() {
    Human me{full_name: "Iozhitsa Sergey", year_of_birth: 2007};
    me.print_info();
    cout << "Age: " << me.get_age(year: 2024) << endl;

    cout << "-----" << endl;
    const auto me_student = new Student(full_name: "Iozhitsa Sergey", year_of_birth: 2007,
                                         admission_to_university_year: 2024, university_name: "BMSTU");
    me_student->print_info();
    cout << "Admission age: " << me_student->get_admission_age() << endl;
    cout << "Age when I end BMSTU: " << me_student->get_age(year: 2030) << endl;

    return 0;
}
```

Рисунок 5 — текст основной (тестирующей) программы

Результат выполнения программы: результат выполнения программы изображен на рисунке 6

```
Human: Iozhitsa Sergey
Year_of_birth: 2007
University name: BMSTU
Admission to university year: 2024
Admission age: 17
Age when I end BMSTU: 23
```

Рисунок 6 — Результат выполнения программы

Вывод: Результатом выполнения работы является иерархия классов, успешно использующая наследование для расширения функциональности базового класса "Human" в производном классе "Student".