**Практическое занятие № 4**

**Тема: Одиночное наследование**

**Задание 1.** Создать проект на основепредложенного текста, **протестировать** его. Рассмотреть  инициализацию объектов производного класса.

Заголовочный файл **Date.h:**

#pragma once

#include <iostream>

#include <string>

class Date

{

private:

int day;

int month;

int year;

public:

Date();

Date(int, int, int);

void Print();

};

Исполняемый файл **Date.cpp:**

#include "Date.h"

Date::Date()

{

day = 0;

month = 0;

year = 0;

}

Date::Date(int day, int month, int year)

{

this->day = day;

this->month = month;

this->year = year;

}

void Date::Print()

{

std::cout << day << ":" << month << ":" << year << std::endl;

}

Заголовочный файл **Birthday.h:**

#pragma once

#include "Date.h"

using std::string;

class Birthday : public Date

{

private:

std::string name;

public:

Birthday();

Birthday(int, int, int, string);

void Print1();

};

Исполняемый файл **Birthday.cpp:**

#include "Birthday.h"

Birthday::Birthday()

{

}

Birthday::Birthday(int day, int month, int year, string name)

:Date(day, month,year)

{

this->name = name;

}

void Birthday::Print1()

{

this->Print();

std::cout << name << std::endl;

}

Исполняемый файл **Main.cpp:**

#include <iostream>

#include "Birthday.h"

int main()

{

Birthday n, n1(10, 12, 2020, "Ann");

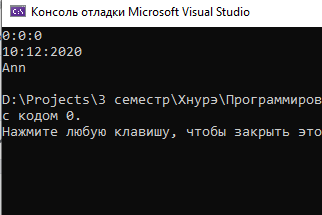
n.Print();

n1.Print1();

return 0;

}

Результат работы программы:



**Задание 2**. Изменить в предложенном тексте спецификатор доступа к данным в базовом классе на **protected**. Внести необходимые изменения в проект и протестировать.

Для изменения доступа к полям базового класса нужно дописать protected перед объявлением переменной.

Заголовочный файл **Date.h:**

#pragma once

#include <iostream>

#include <string>

class Date

{

protected: // Спецификатор доступа

int day;

int month;

int year;

public:

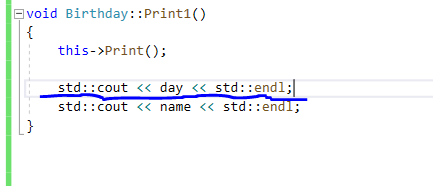
Date();

Date(int, int, int);

void Print();

};

Теперь к полям класса Date могут обращаться дочерние классы



**Задание 3**. Внести изменения в данный проект согласно выбранного варианта задания. и **протестировать** проект. Спецификатор доступа к данным в базовом классе установить **private.**

11. В производном классе разработать метод сравнения даты рождения двух человек и вернуть информацию о младшем. Параметр передать по указателю и результат метода передать по значению

Заголовочный файл **Date.h:**

#pragma once

#include <iostream>

#include <string>

class Date

{

private: // Спецификатор доступа

int day;

int month;

int year;

public:

Date();

Date(int, int, int);

void Print();

void GetDate(int& day, int& month, int& year);

};

Исполняемый файл **Date.cpp:**

#include "Date.h"

Date::Date()

{

day = 0;

month = 0;

year = 0;

}

Date::Date(int day, int month, int year)

{

this->day = day;

this->month = month;

this->year = year;

}

void Date::Print()

{

std::cout << day << ":" << month << ":" << year << std::endl;

}

void Date::GetDate(int& day, int& month, int& year)

{

day = this->day;

month = this->month;

year = this->year;

}

Заголовочный файл **Birthday.h:**

#pragma once

#include "Date.h"

using std::string;

class Birthday : public Date

{

private:

std::string name;

public:

Birthday();

Birthday(int, int, int, string);

void Print1();

Birthday Equals(Birthday& other);

string GetName();

};

Исполняемый файл **Birthday.cpp:**

#include "Birthday.h"

Birthday::Birthday()

{

}

Birthday::Birthday(int day, int month, int year, string name)

:Date(day, month,year)

{

this->name = name;

}

void Birthday::Print1()

{

this->Print();

std::cout << name << std::endl;

}

string Birthday::GetName()

{

return name;

}

Birthday Birthday::Equals(Birthday& other)

{

int thisYear, thisMonth, thisDay;

int otherYear, otherMonth, otherDay;

this->GetDate(thisDay, thisMonth, thisYear);

other.GetDate(otherDay, otherMonth, otherYear);

if (thisYear < otherYear || (thisYear == otherYear && thisMonth < otherMonth) || (thisYear == otherYear && thisMonth == otherMonth && thisDay < otherDay))

{

return Birthday(thisDay, thisMonth, thisYear, name);

}

else

{

return Birthday(otherDay, otherMonth, otherYear, other.GetName());

}

}

Исполняемый файл **Main.cpp:**

#include <iostream>

#include "Birthday.h"

int main()

{

Birthday n(9, 12, 2020, "Ivan"), n1(10, 10, 2020, "Ann");

Birthday result = n.Equals(n1);

result.Print1();

system("pause");

return 0;

}

Пример работы программы:

