Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

(ПНИПУ)

**Факультет:** электротехнический

**Направление:** Промышленная мехатроника и робототехника

ОТЧЁТ

По лабораторной работе №5

«Классы»

Семестр 1

Выполнил:

студент группы ПРТ-21-1б

Поздняков Е.С.

Пермь 2022

**Задание №1**

**Постановка задачи**

Базовый класс:

ТРОЙКА ЧИСЕЛ (Triad)

First – int

Second – int

Third – int

Определить методы изменения полей и увеличения полей на 1.

Создать производный класс DATE с полями год, месяц и число. Переопределить методы увеличения полей на 1 и определить метод увеличения даты на n дней.

**Анализ**

1. Пользователь вводит значение длин двух катетов.
2. Создается экземпляр класса A, класс А содержит поля first и second – значения двух катетов, функции инициализации переменных и функцию hypotenuse, которая возвращает значение гипотенузы по заданных значениям катетов.
3. При помощи функций SetFirst и SetSecond устанавливаются значения длин катетов. Далее переменной h присваивается результат работы функции hypotenuse.
4. Полученный результат выводится.

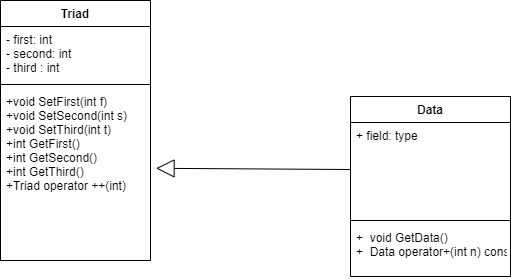
**Используемые переменные**

int n – количество дней;

Triad m1 – тройка чисел;

Data d1 – дата;

**UML-диаграмма**



**Код C++**

#include <iostream>

using namespace std;

class Data;

class Triad

{

protected:

int first;

int second;

int third;

public:

Triad() : first(0), second(0), third() {}

Triad(int f, int s, int t) : first(f), second(s), third(t) {}

Triad(const Triad &t)

{

first = t.first;

second = t.second;

third = t.third;

}

~Triad(){}

void SetFirst(int f){first = f;}

void SetSecond(int s){second = s;}

void SetThird(int t){third = t;}

int GetFirst(){return first;}

int GetSecond(){return second;}

int GetThird(){return third;}

Triad operator +()

{

Triad t;

t.first = this->first + 1;

t.second = this->second + 1;

t.third = this->third + 1;

return t;

}

Triad operator ++(int)

{

Triad t;

t.first = this->first + 1;

t.second = this->second + 1;

t.third = this->third + 1;

return t;

}

friend istream& operator >> (istream&in, Triad&t);

friend ostream& operator << (ostream&out, const Triad&t);

};

istream& operator>>(istream&in, Triad&t)

{

cout << "Input first: ";

in >> t.first;

cout << "Input second: ";

in >> t.second;

cout << "Input third: ";

in >> t.third;

return in;

}

ostream& operator<<(ostream&out, const Triad&t)

{

return (out << "{" << t.first << ", " << t.second << ", " << t.third << "}");

}

class Data : public Triad

{

public:

Data() : Triad() {}

Data(int f, int s, int t) : Triad(f, s, t)

{}

Data(const Data &t)

{

first = t.first;

second = t.second;

third = t.third;

}

~Data(){}

void GetData()

{

cout << first << "."<< second << "." << third << endl;

}

Data operator+(int n) const

{

Data t;

t.first = this->first + n;

if (t.first > 31)

{

t.first = this->first + n - 31;

t.second = this->second + 1;

if (t.second > 12)

{

t.second = 1;

t.third = this->third + 1;

}

else

{

t.third = this->third;

}

}

else

{

t.second = this->second;

t.third = this->third;

}

if (t.first > 31)

{

t.first = t.first - 31;

t.second++;

}

return t;

}

};

int main()

{

int n;

Triad m1(1, 2, 2003);

cout << "m1 = " << m1 << endl;

m1 = m1++;

cout << "m1 + 1 = " << m1 << endl;

Data d1(28, 12, 2003);

cout << "d1 = ";

d1.GetData();

cout << "Input amount of days: ";

cin >> n;

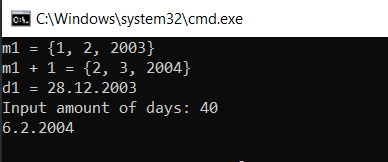
d1 = d1 + n;

d1.GetData();

return 0;

}

**Скриншоты результатов**

****

**Анализ результатов**

Изначально заданы значения тройки чисел. К каждому числу прибавляется 1. Полученная тройка выводится на экран. Выводится значение даты d1. Пользователь вводит количество дней, которые нужно прибавить к дате. Полученный результат выводится на экран.

**Контрольные вопросы**

1. Какой метод называется чисто виртуальным? Чем он отличается от виртуального метода?

Чисто виртуальный метод содержит признак = 0 вместо тела, например:

virtual void f(int) = 0;

Чисто виртуальный метод должен переопределяться в производном классе.

Виртуальная функция – это функция, которая определяется в базовом классе, но любой производный класс может ее переопределить

2. Какой класс называется абстрактным?

Абстрактным называется класс, содержащий хотя бы один чисто виртуальный метод.

3. Для чего предназначены абстрактные классы?

Предназначены для представления общих понятий, которые нужно конкретизировать в производных классах.

4. Что такое полиморфные функции?

Полиморфные функции – это функции, которые могут вызываться с аргументами различного типа, а фактически выполняемый код зависит от типа аргумента

5. Чем полиморфизм отличается от принципа подстановки?

Полиморфизм – это способность программы обрабатывать данные разных типов одним кодом, а принцип подстановки заключается в том, что, где можно использовать производный класс, там можно использовать и базовый.

6. Привести примеры иерархий с использованием абстрактных классов.

Абстрактный класс – фигура, в нем объявлена чисто виртуальная функция для вывода параметров фигуры. Производные классы – треугольник и круг, в каждом из них реализована функция для вывода параметров. В зависимости от вида фигуры будет выполняться та или иная функция.

7. Привести примеры полиморфных функций.

Существует базовый класс человек, он содержит поля имя и возраст, в нем определена виртуальная функция для вывода этих параметров. Существует производный класс Работник, в котором определяется еще одно поле – зарплата. Для вывода полной информации о работнике следует переопределить функцию для вывода параметров.

8. В каких случаях используется механизм позднего связывания?

Позднее связывание нужно, если требуется реализовать полиморфизм.