Задача. Створити клас TVSeries, який представляє прогресію і містить методи для знаходження -го члена прогресії і знаходження суми перших  членів цієї прогресі. На основі цього класу створити класи нащадки, які представляють арифметичні та геометричні прогресії. Випадковим чином згенерувати дані для  прогресій (геометрична, арифметична, геометрична, арифметична, і т.д.). Знайти суму перших  членів прогресії, ‑товий член якої є найбільшим.

#pragma once

class TVSeries

{

protected:

double a1, d;

public:

virtual double get\_an(int n)=0;

virtual double get\_Sn(int n) = 0;

TVSeries(double a1, double d);

TVSeries();

};

#include "stdafx.h"

#include "TVSeries.h"

TVSeries::TVSeries(double a1, double d)

{

this->a1 = a1;

this->d = d;

}

TVSeries::TVSeries():TVSeries(0,0)

{

}

#pragma once

#include "TVSeries.h"

class ArithmethicSeries :

public TVSeries

{

public:

ArithmethicSeries(double a1, double d);

double get\_an(int n);

double get\_Sn(int n);

};

#include "stdafx.h"

#include "ArithmethicSeries.h"

ArithmethicSeries::ArithmethicSeries(double a1, double d):TVSeries(a1,d)

{

}

double ArithmethicSeries::get\_an(int n)

{

return a1+(n-1)\*d;

}

double ArithmethicSeries::get\_Sn(int n)

{

return (2\*a1+d\*(n-1))/2\*n;

}

#pragma once

#include "TVSeries.h"

class GeomSerioes :

public TVSeries

{

public:

GeomSerioes(double a1, double d);

GeomSerioes();

double get\_an(int n);

double get\_Sn(int n);

};

#include "stdafx.h"

#include "GeomSerioes.h"

#include <math.h>

GeomSerioes::GeomSerioes(double a1, double d):TVSeries(a1,d)

{

}

GeomSerioes::GeomSerioes():TVSeries(0,0)

{

}

double GeomSerioes::get\_an(int n)

{

return a1\*pow(d,n-1);

}

double GeomSerioes::get\_Sn(int n)

{

return a1\*(pow(d,n)-1)/(d-1);

}

// ConsoleApplication36.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "TVSeries.h"

#include "ArithmethicSeries.h"

#include "GeomSerioes.h"

using namespace std;

int main()

{

int seriesCount;

printf("Series count=");

cin >> seriesCount;

// Створюємо масив покажчиків (будемо зберігати адреси нащадків)

TVSeries\* \* series = new TVSeries\*[seriesCount];

//Створюємо динамічно seriesCount об"єктів нащадків

for (int i = 0; i < seriesCount; i++)

{

double a1 = rand() % 100; //Випадковим чином визначаємо 1 ел.

double d = rand() % 100; //і степінь (крок)прогресії

if (rand()%2==0) //Випадковим чином визначаємо яку прогр. створити (0-арифм., 1-геом.)

{

series[i] = new ArithmethicSeries(a1,d);

}

else

{

series[i] = new GeomSerioes(a1, d);

}

}

//-----------

printf("n=");

int n;

cin >> n;

printf("m=");

int m;

cin >> m;

//--- Знаходимо прогресію, у якщо найбільший n-вий елемент

double max = series[0]->get\_an(n);

int max\_index = 0;

for (int i = 1; i < seriesCount; i++)

{

double temp= series[i]->get\_an(n);

if (max<temp)

{

max = temp;

max\_index = i;

}

}

//---------

double sum = series[max\_index]->get\_Sn(m);

printf("Sum=%f",sum);

system("pause");

return 0;

}