1 ЦЕЛЬ РАБОТЫ

Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизмов наследования и виртуальных функций. Освоение особенностей отладки объектно-ориентированных программ.

2 ПОСТАНОВКА ЗАДАЧИ

Для заданной по варианту иерархии описать классы, конструкторы и деструктор, функции ввода и вывода информации на экран. Базовый класс определить как абстрактный, а заданную функцию — как чисто виртуальную в базовом классе и переопределить ее в остальных классах иерархии. Проиллюстрировать корректную работу виртуальных функций и механизма наследования.

Вариант 5

Создать абстрактный базовый класс Прогрессия с виртуальной функцией Сумма прогрессии. Создать производные классы: арифметическая прогрессия и геометрическая прогрессия. Каждый класс имеет два поля типа double. Первое — первый член прогрессии, второе — постоянная разность (для арифметической) и постоянное отношение (для геометрической). Определить функцию вычисления суммы, где параметром является количество элементов прогрессии.

Арифметическая прогрессия:

Сумма арифметической прогрессии:

Геометрическая прогрессия:

Сумма геометрической прогрессии:

3 ТЕКСТ ПРОГРАММЫ

#include <iostream>

#include <cmath>

using namespace std;

class Progress

{

public:

double \*a;

int len;

Progress(int n)

{

a=new double[n];

len=n;

}

virtual ~Progress()

{

delete[]a;

}

virtual double SumPr() = 0;

};

class ArPr : public Progress

{

private:

double fa, d;

public:

ArPr(int n, double \_fa, double \_d) : Progress(n)

{

fa=\_fa; d=\_d;

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

{

a[i]=fa+i\*d;

cout<<a[i]<<", ";

}

cout<<endl;

}

virtual double SumPr()

{

double s=0;

for(int i=0; i<len; i++) s+=(len+1)\*(fa+a[i])/2;

return s;

}

};

class GeoPr : public Progress

{

private:

double fg, r;

public:

GeoPr(int n, double \_fg, double \_r) : Progress(n)

{

fg=\_fg; r=\_r;

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

{

a[i]=fg\*pow(r,i);

cout<<a[i]<<", ";

}

cout<<endl;

}

virtual double SumPr()

{

double s=0;

for(int i=0; i<len; i++) s+=(fg-a[i]\*r)/(1-r);

return s;

}

};

int main ()

{

double fa, d, fg, r, n;

setlocale(0, "rus");

cout<<"Введите размер: ";

cin>>n;

cout<<"Введите начальное значение: ";

cin>>fa;

cout<<"Введите постоянную разность: ";

cin>>d;

Progress \*ob = new ArPr(n, fa, d);

cout<<"Сумма арифм. прогрессии: "<<ob->SumPr()<<endl;

delete ob;

cout<<"--------------------------------------"<<endl;

cout<<"Введите начальное значение: ";

cin>>fg;

cout<<"Введите постоянное отношение: ";

cin>>r;

ob = new GeoPr(n, fg, r);

cout<<"Сумма геом. прогрессии: "<<ob->SumPr()<<endl;

delete ob;

}

4 ТЕСТИРОВАНИЕ

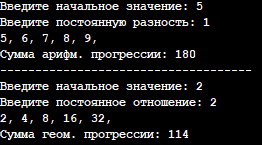


Рисунок 1 – Тест №1 с 5 элементами

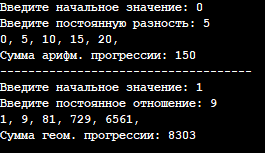


Рисунок 2 – Тест №2 с 5 элементами

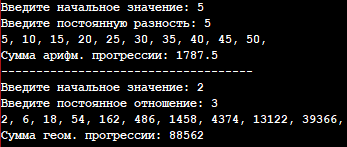


Рисунок 3 – Тест с 10 элементами

ВЫВОД

В ходе выполнения лабораторной работы были получены навыки работы с механизмом наследованием. Изучены спецификаторы доступа в производном классе, виртуальные методы и абстрактные классы. Была написана программа, описывающая базовый класс как абстрактный, а заданную функцию — как чисто виртуальную в базовом классе и переопределяющая ее в остальных классах иерархии. Была проиллюстрирована корректная работа виртуальных функций и механизма наследования.