1 ЦЕЛЬ РАБОТЫ

Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизма множественного наследования.

2 ПОСТАНОВКА ЗАДАЧИ

Описать интерфейс ввода-вывода. Описать иерархию классов, заданную по варианту. Для каждого класса описать конструкторы и деструктор (по необходимости), функции ввода и вывода значений полей. В каждом классе должны присутствовать минимум одно уникальное поле и один уникальный метод. Проиллюстрировать корректную работу механизма множественного наследования — для этого создать объекты базовых классов и заполнить их поля данными, вывести на печать. Создать объект класса-наследника, его поля заполнить значениями соответствующих полей базовых классов. Вывести на печать данные полученного объекта.

Вариант 5

Базовые классы: Вода (объем), Вещество (вес, название, тип). Класс-наследник: Раствор.

3 ТЕКСТ ПРОГРАММЫ

#include <conio.h>

#include <stdlib.h>

#include <iostream>

using namespace std;

class Aqua

{

public:

int obm;

Aqua()

{

cout<<"Введите объём воды: ";

cin>>obm;

}

void Vivod()

{

cout<<"Данные класса Вода:"<<endl;

cout<<"Объём воды: "<<obm<<endl;

}

};

class Substance

{

public:

int ves;

char \*name, \*type;

Substance()

{

name=new char [10];

type=new char [10];

cout<<"Введите вес вещества: ";

cin>>ves;

cout<<" Введите имя вещества: ";

cin>>name;

cout<<" Введите тип вещества: ";

cin>>type;

}

~Substance()

{

delete [] name;

delete [] type;

}

void Vivod()

{

cout<<"Данные класса Вещество:"<<endl;

cout<<"Вес вещества: "<<ves<<endl;

cout<<"Имя вещества: "<<name<<endl;

cout<<"Тип вещества: "<<type<<endl;

}

};

class Rastvor : public Aqua, public Substance

{

public:

int plus, minus;

void Ink()

{

cout<<"Введите на сколько увеличить объём: "<<endl;

cin>>plus;

obm+=plus;

cout<<"Введите на сколько уменьшить вес: "<<endl;

cin>>minus;

ves-=minus;

cout<<"Âûïîëíåíî!"<<endl;

}

void Vivod()

{

cout<<"Объём воды: " <<obm<<endl;

cout<<"Вес вещества: "<<ves<<endl;

cout<<" Имя вещества: "<<name<<endl;

cout<<" Тип вещества: "<<type<<endl;

}

};

int main ()

{

setlocale(0, "RUS");

int key;

cout<<" Создание объектов базового класса:"<<endl;

Rastvor ob3;

while (1)

{

system("cls");

cout << "1 - Вызов объектом производного класса методов базовых классов(Просмотр)" << endl;

cout << "2 - Просмотр объекта производного класса " << endl;

cout << "3 - Вызов уникального метода производного класса " << endl;

cout << "4 - Выход " << endl;

key=getch();

switch (key)

{

case'1':

{

ob3.Aqua::Vivod();

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

ob3.Substance::Vivod();

system("pause");

break;

}

case '2':

{

cout<<" Данные класса Раствор:"<<endl;

ob3.Vivod();

system("pause");

break;

}

case '3':

{

ob3.Ink();

system("pause");

break;

}

case '4': return 0;

default: cout << " Введите число от 1 до 4: " << endl;

}

}

}

4 ТЕСТИРОВАНИЕ

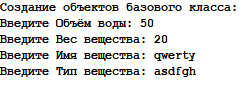


Рисунок 1 – Создание объектов базового класса

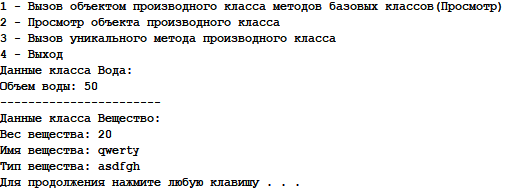


Рисунок 2 – Просмотр данных базового класса

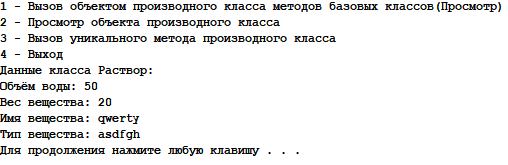


Рисунок 3 – Просмотр данных производного класса

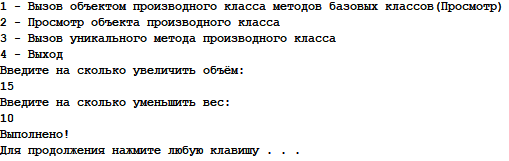


Рисунок 4 – Уникальный метод производного класса

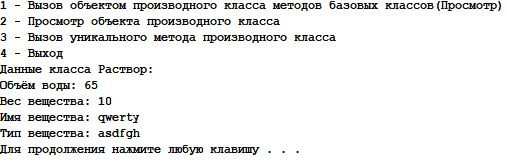


Рисунок 5 – Данные после использования уникального метода

ВЫВОД

В ходе выполнения лабораторной работы были получены навыки работы с использованием механизма множественного наследования. Изучены основные термины и структура множественного наследования. Была написана программа, описывающая иерархию классов, заданную по варианту. Проиллюстрирована корректная работа механизма множественного наследования – созданы объекты базовых классов и заполнены полями, выведены на печать. Был создан объект класса-наследника, его поля заполнены значениями базовых классов. Выведены на печать данные полученного объекта.