Лабораторная работа 5

Одиночное наследование

Цель работы:

Изучить способы создания производного класса и особенности работы с ним, правила инициализации и доступа к элементам производного класса; приобрести практические навыки наследования.

Задание:

Создайте производный класс для АТД, реализованного по заданию лабораторной работы 4, используя одиночное наследование. Проверьте работоспособность АТД на тестовом наборе данных.

#include <iostream>

#include <string>

#include <clocale>

using namespace std;

class Meow1

{

private:

int number;

int volume;

char\* color;

bool tempreg;

public:

Meow1();

Meow1(int a, int b, char\* c, bool d);

Meow1(const Meow1& a);

~Meow1();

void Print() const {

cout << "Серийный номер: " << number << endl;

cout << "Объём: " << volume << endl;

cout << "Регулятор температуры ";

if (tempreg)

{

cout << "имеется";

}

else

{

cout << "отсутствует";

}

cout << endl;

cout << "Цвет: " << color << endl;

};

Meow1 sum(const Meow1& a) {

Meow1 f;

char\* tmp = new char[strlen(color) + 1];

strcpy(tmp, color);

f.number = number + a.number;

f.volume = volume + a.volume;

f.tempreg = tempreg + a.tempreg;

strcat(tmp, "-");

strcat(tmp, a.color);

f.color = new char[strlen(tmp) + 1];

strcpy(f.color, tmp);

return f;

};

bool Cmp(const Meow1& a) {

return volume == a.volume && number == a.number && tempreg == a.tempreg && color == a.color;

};

void Copy(const Meow1& a) {

number = a.number;

volume = a.volume;

color = new char[strlen(a.color) + 1];

strcpy(color, a.color);

tempreg = a.tempreg;

}

};

class Meow2 : public Meow1

{

private:

int size;

public:

Meow2();

Meow2(int a, int b, char\* c, bool d, int s);

~Meow2();

void Print() const;

};

Meow1::Meow1()

{

number = 0;

volume = 0;

tempreg = false;

color = new char[1];

strcpy(color, "white");

}

Meow2::Meow2() :Meow1()

{

size = 0;

}

Meow1::Meow1(int a, int b, char\* c, bool d)

{

number = a;

volume = b;

tempreg = d;

color = new char[strlen(c) + 1];

strcpy(color, c);

}

Meow2::Meow2(int a, int b, char\* c, bool d, int s) :Meow1(a, b, c, d )

{

size = s;

}

Meow1::Meow1(const Meow1& a)

{

if (&a != this)

{

number = a.number;

volume = a.volume;

color = new char[strlen(a.color) + 1];

strcpy(color, a.color);

tempreg = a.tempreg;

}

}

Meow1 :: ~Meow1()

{

delete[] color;

}

Meow2 :: ~Meow2()

{

Meow1::~Meow1();

}

void Meow2::Print() const

{

Meow1::Print();

cout << "Размер нагревательного прибора: " << size << endl;

}

int main()

{

setlocale(LC\_ALL, "rus");

Meow2 mur1, mur2(1, 1, (char\*)"black", false, 15), mur3(100, 1223, (char\*)"white", true, 20);

cout << "Информация о нагревательных элементах: " << endl;

mur1.Print();

mur2.Print();

mur3.Print();

system("pause");

return 0;

}

