Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №1**

**По дисциплине ЯП за 3 семестр**  
**Тема: «**Классы и объекты C++»

**Выполнил:**

Студент группы ПО-6(1)  
 2-го курса

Мартынович Даниил

**Проверила:**

Хацкевич М. В.

Брест 2021

Задание. Вариант 13

В соответствии с вариантом 13, требуется реализовать класс «страны», который содержит 3 переменных (имя – char\*, форма правления – char\*, площадь – float).

1.Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

2.Определить в классе деструктор.

3.Определить в классе компоненты-функции для просмотра и установки полей данных.

4.Определить в классе указатель на компоненту-функции.

5.Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора или деструктора сопровождается выдачей соответствующего сообщения. Какой объект какой конструктор вызвал.

Ход работы

Пример определения класса.

class Country

{

private:

char\* name; //Имя

char\* info; //Форма правления

float size; //площадь

public:

Country();//Конструктор без параметров

Country(const char newname[], const char newinfo[], float newsize);//Конструторк с параметрами

Country(const Country&other);//Конструктор копирования

~Country();//Деструктор

//Компонентные функции

//Вывод

char\* Getname();

char\* Getinfo();

float Getsize();

//Ввод

void Setname(const char\* newname);

void Setinfo(const char\* newinfo);

void Setsize(float newsize);

void Set(const char\* newname, const char\* newinfo, float newsize);

void Print();

};

1. Реализация конструкторов

Конструктор без параметров:

Country::Country()

{

name = (char\*)calloc(15, sizeof(char));

strcpy(name, "None");

info = (char\*)calloc(15, sizeof(char));

strcpy(info, "None");

size = 189,2;

cout << "\nКонструктор без параметров вызван для объекта"<< this << endl;

}

Конструктор с параметрами:

Country::Country(const char newname[], const char newinfo[], float newsize)

{

name = (char\*) calloc(strlen(newname),sizeof(char));

strcpy(name, newname);

info = (char\*)calloc(strlen(newinfo), sizeof(char));

strcpy(info, newinfo);

size = newsize;

cout << "\nКонструторк с параметрами вызван для объекта" << this << endl;

}

Конструктор копирования:

Country::Country(const Country&other)

{

name = (char\*)calloc(strlen(other.name), sizeof(char));

strcpy(name, other.name);

info = (char\*)calloc(strlen(other.info), sizeof(char));

strcpy(info, other.info);

size = other.size;

cout << "\nКонструторк копирования вызван для объекта" << this << endl;

}

1. Определение деструктора

Country::~Country()

{

free(name);

free(info);

cout << "Вызван деструктор для обьекта" << this << endl;

}

1. Определение в классе компонентных-функций для просмотра и установки полей данных:

Для просмотра полей:

char\* Country::Getname()

{

return name;

}

char\* Country::Getinfo()

{

return info;

}

float Country::Getsize()

{

return size;

}

Для установки полей:

void Country::Setname(const char\* newname)

{

free(name);

name = (char\*)calloc(strlen(newname), sizeof(char));

strcpy(name, newname);

}

void Country::Setinfo(const char\* newinfo)

{

free(info);

info = (char\*)calloc(strlen(newinfo), sizeof(char));

strcpy(info, newinfo);

}

void Country::Setsize(float newsize)

{

size = newsize;

}

void Country::Set(const char\* newname, const char\* newinfo, float newsize)

{

Setname(newname);

Setinfo(newinfo);

Setsize(newsize);

}

void Country::Print()

{

cout << "Country name: " << this->name << "\t System: " << this->info << "\t Size: " << this->size << endl;

}

1. Определение в классе указателя на компоненту-функции:

void (Country:: \* pf)();//Указатель на компонентную функцию void Print();

pf = &Country::Print;

(l.\*pf)();

5. В программе используется три файла:

Заголовочный h-файл с определением класса,

Cpp-файл с реализацией класса,

Cpp-файл демонстрационной программы

Заголовочный h-файл lib.h

# ifndef LIB\_H

# define LIB\_H

# include < cstring >

# include < cstdio >

# include < cstdlib >

#include <iostream>

using namespace std;

class Country

{

private:

char\* name; //Имя

char\* info; //Форма правления

float size; //площадь

public:

Country();//Конструктор без параметров

Country(const char newname[], const char newinfo[], float newsize);//Конструторк с параметрами

Country(const Country&other);//Конструктор копирования

~Country();//Деструктор

//Компонентные функции

//Вывод

char\* Getname();

char\* Getinfo();

float Getsize();

//Ввод

void Setname(const char\* newname);

void Setinfo(const char\* newinfo);

void Setsize(float newsize);

void Set(const char\* newname, const char\* newinfo, float newsize);

void Print();

};

# endif

Cpp-файл с реализацией класса lib realization.cpp

#include "lib.h"

using namespace std;

//Геттеры

char\* Country::Getname()

{

return name;

}

char\* Country::Getinfo()

{

return info;

}

float Country::Getsize()

{

return size;

}

//Сеттеры

void Country::Setname(const char\* newname)

{

free(name);

name = (char\*)calloc(strlen(newname), sizeof(char));

strcpy(name, newname);

}

void Country::Setinfo(const char\* newinfo)

{

free(info);

info = (char\*)calloc(strlen(newinfo), sizeof(char));

strcpy(info, newinfo);

}

void Country::Setsize(float newsize)

{

size = newsize;

}

void Country::Set(const char\* newname, const char\* newinfo, float newsize)

{

Setname(newname);

Setinfo(newinfo);

Setsize(newsize);

}

void Country::Print()

{

cout << "Country name: " << this->name << "\t System: " << this->info << "\t Size: " << this->size << endl;

}

//Конструкторы

Country::Country()

{

name = (char\*)calloc(15, sizeof(char));

strcpy(name, "None");

info = (char\*)calloc(15, sizeof(char));

strcpy(info, "None");

size = 189,2;

cout << "\nКонструктор без параметров вызван для объекта"<< this << endl;

}

Country::Country(const char newname[], const char newinfo[], float newsize)

{

name = (char\*) calloc(strlen(newname),sizeof(char));

strcpy(name, newname);

info = (char\*)calloc(strlen(newinfo), sizeof(char));

strcpy(info, newinfo);

size = newsize;

cout << "\nКонструторк с параметрами вызван для объекта" << this << endl;

}

Country::Country(const Country&other)

{

name = (char\*)calloc(strlen(other.name), sizeof(char));

strcpy(name, other.name);

info = (char\*)calloc(strlen(other.info), sizeof(char));

strcpy(info, other.info);

size = other.size;

cout << "\nКонструторк копирования вызван для объекта" << this << endl;

}

//Деструктор

Country::~Country()

{

free(name);

free(info);

cout << "Вызван деструктор для обьекта" << this << endl;

}

Cpp-файл демонстрационной программы lib demonstration.cpp

#include "lib.h"

int main()

{

setlocale(LC\_ALL, "RU");

Country a;//Без параметров

a.Print();

Country b("India","vostok",123);//с параметрами

b.Print();

//массив студентов размещается в статической памяти

Country c[3];

c[0].Set("DANYA","DEPIS", 50);

c[1].Set("Moldova", "RIP", 120);

c[2].Set("Poland", "UK", 12);

c[0].Print();

c[1].Print();

c[2].Print();

//массив студентов размещается в динамической памяти

Country\* p;

p = new Country[3];

p->Set("France","Departament",12);

p->Set("Kingdom", "Fabio", 19);

p->Set("Mokedoni", "Large", 48);

p[0].Print();

p[1].Print();

p[2].Print();

Country l;

l.Set("England", "Monarch", 172);

void (Country:: \* pf)();//Указатель на компонентную функцию

pf = &Country::Print;

(l.\*pf)();

system("pause");

}

Вывод: научился использовать классы и объекты в C++. Реализовал класс согласно варианту. Определил конструкторы и деструкторы, компоненты-функций для просмотра и установки полей данных.