

Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №2

По дисциплине “Языки программирования”

Вариант №7

Выполнил:

Кравцевич Г.А. (ПО-7,2)

Проверил:

Дряпко. А. В.

Дата выполнения:

14.10.21

Брест 2021

Цель:

Получить практические навыки создания иерархии классов и использования статических компонентов класса.

Задание:

1. Определить иерархию классов (в соответствии с вариантом).
2. Определить в классе статическую компоненту - указатель на начало связанного списка объектов и статическую функцию для просмотра списка.
3. Реализовать классы.
4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается.
5. Сделать соответствующие методы не виртуальными и посмотреть, что будет.
6. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

Иерархия классов:

Тест, экзамен, выпускной экзамен, испытание

Код программы:**Файл Test.h:**

```
#pragma once
#include <list>
#include <iostream>
#include <string>

using namespace std;

class Test
{
private:
    static list<Test*> _list;

protected:
```

```

        string _name;
        int _complexity;

public:
    Test(int, string);

    virtual ~Test();
    virtual void Add() = 0;
    virtual void Show() = 0;

    static void ShowAll();
};

```

Файл Test.cpp:

```

#include "Test.h"
#include <list>

list<Test*> Test::_list;

Test::Test(int complexity = 0, string name = "")
{
    std::cout << "Test constructor" << endl;

    this->_complexity = complexity;
    this->_name = name;
}

Test::~Test()
{
    std::cout << "Test destructor" << endl;
}

```

Файл Exam.h:

```

#pragma once
#include "Test.h"

```

```

#include <list>

class Exam :
    public Test
{
private:
    static list<Exam*> _list;

protected:
    string _subject;

public:
    Exam(int compecity, string name, string subject);

    ~Exam() override;
    void Show() override;
    void Add() override;

    static void ShowAll();
};

```

Файл Exam.cpp:

```

#include "Exam.h"

list<Exam*> Exam::_list;

// Constructors
Exam::Exam(int complexity = 0, string name = "", string subject = "") :
Test(complexity, name)
{
    cout << "Exam constructor" << endl;

    this->_subject = subject;
    this->Add();
}

```

```

// Destructor
Exam::~Exam()
{
    cout << "Exam destructor" << endl;

    Test::~Test();
}

// Methods
void Exam::Show()
{
    cout << endl;
    cout << "Type: Exam" << endl;
    cout << "Name:" << this->_name << endl;
    cout << "Complexity: " << this->_complexity << endl;
    cout << "Subject: " << this->_subject << endl;
}

void Exam::Add()
{
    _list.push_back(this);
}

void Exam::ShowAll()
{
    for (Exam* e : Exam::_list)
    {
        e->Show();
    }
}

```

Файл FinalExam.h:

```

#pragma once
#include "Exam.h"

```

```

class FinalExam :
    public Exam
{
private:
    static list<FinalExam*> _list;

protected:
    int _mark;

public:
    FinalExam(int comp, string name, string sybj, int mark);

    ~FinalExam() override;
    void Add() override;
    void Show() override;

    static void ShowAll();
};

```

Файл FinalExam.cpp:

```

#include "FinalExam.h"

```

```

list <FinalExam*> FinalExam::_list;

```

```

// Constructors

```

```

FinalExam::FinalExam(int complexity = 0, string name = "", string subject = "",
int mark = 0) : Exam(complexity, name, subject)
{
    cout << "FinalExam constructor" << endl;

    this->_mark = mark;
    this->Add();
}

```

```

// Destructor

```

```

FinalExam::~~FinalExam()

```

```

{
    cout << "FinalExam destructor" << endl;

    Exam::~~Exam();
}

// Methods
void FinalExam::Show()
{
    cout << endl;
    cout << "Type: FinalExam" << endl;
    Exam::Show();
    cout << "Mark: " << this->_mark << endl;
}

void FinalExam::Add()
{
    _list.push_back(this);
}

void FinalExam::ShowAll()
{
    for (FinalExam* fe : FinalExam::_list)
    {
        fe->Show();
    }
}

```

Файл main.cpp:

```

#include <iostream>
#include "Exam.h"
#include "FinalExam.h"
#include "Trial.h"

int main()

```

```

{
    Exam e(1, "name", "subject");
    Exam e2(2, "name2", "subject2");

    FinalExam fe(1, "fname", "fsubject", 10);
    FinalExam fe2(2, "fname2", "fsubject2", 12);

    Trial t(1, "tname", "sponsor");
    Trial t2(2, "tname2", "sponsor2");

    Exam::ShowAll();
    FinalExam::ShowAll();
    Trial::ShowAll();

    getchar();
}

```

Результат работы программы:

```

D:\Projects\Лабораторные\ЯП\Prog_lang\reports\Кравцевич\2\src\Lab2\Debug\Lab2.exe
Test constructor
Exam constructor
Test constructor
Exam constructor
Test constructor
Exam constructor
FinalExam constructor
Test constructor
Exam constructor
FinalExam constructor
Test constructor
Trial constructor
Test constructor
Trial constructor

Type: Exam
Name: name
Complexity: 1
Subject: subject

Type: Exam
Name: name2
Complexity: 2
Subject: subject2

Type: FinalExam
Name: fname
Complexity: 1

```

Вывод:

Получил практические навыки создания иерархии классов и использования статических компонентов класса.