

Звіт

Автор: Момот Р.Є. КІТ-119а

Дата: 17 лютого 2020

Лабораторна робота №1. Класи

Тема. Класи та специфікатори доступу. Інкапсуляція. Константи.

Мета: отримати базові знання про класи. Дослідити механізм інкапсуляції.

1. Завдання до роботи

Індивідуальне завдання:

Для предметної галузі розробити два класи:

- клас, що відображає сутність «базового класу». При цьому, в даному класі повинно бути мінімум три числових поля (бажано, щоб одне з цих полів було унікальним ідентифікатором об'єкта);

- клас, що має у собі динамічний масив об'єктів базового класу та має в собі методи додавання, видалення елемента, отримання елемента по індексу (або ідентифікатору), вивід усіх елементів на екран.

Прикладна галузь: програмне забезпечення

Базовий клас: програма, що виконується

2. Опис класів, змінних, методів та функцій

2.1 Опис класів

Базовий клас: C_Program

Клас, що має в собі масив базового класу та методи для роботи з ним: C_List

2.2 Опис змінних

`int` TimeOfWork – поле класу Program(час виконання програми).

`int` size – поле класу Program(розмір програми у мегабайтах).

`int` AmountOfLines – поле класу Program(кількість рядків коду).

`int` listSize – поле класу C_List(розмір масиву елементів класу Program).

`C_Program*` List – поле класу C_Program(масив елементів класу Program).

`C_List` List – об'єкт класу C_List.

`C_Program` getProgram, newProgram – змінні для нових програм, необхідні для роботи програми.

2.3 Опис методів

`void setListSize(int)` – запис даних у змінну розміру масиву елементів класу Program (метод класу C_List).
`int getListSize() const` – отримання даних змінної розміру масиву елементів класу Program (метод класу C_List).
`void CreateList()` – створення масиву елементів і заповнення даними (метод класу C_List).
`void PrintAll() const` – виведення даних елементів у консоль (метод класу C_List).
`void PrintOneEl(int) const` – виведення даних одного елементу у консоль (метод класу C_List).
`void AddEl(C_Program&)` – додавання нового елементу в масив (метод класу C_List).
`void DeleteEl(int)` – видалення елемента з масиву (метод класу C_List).
`void FreeMemory()` – звільнення динамічного масиву (метод класу C_List).
`C_Program GetProgramID(int) const` – отримання даних елемента по індексу (метод класу C_List).

2.4 Опис функцій

`void Menu()` – функція меню.
`void Test_GetProgramID(C_List, int&)` – тест функції знаходження та повернення об'єкту по індексу.
`void Test_AddEl(C_List&, int&)` – тест функції додавання об'єкта до масиву об'єктів.
`void Test_DelEl(C_List&, int&)` – тест функції видалення об'єкта з масиву об'єктів.

3. Текст програми

Test.cpp

```
#include "List.h"
#include "Program.h"

void Test_GetProgramID(C_List, int&);
void Test_AddEl(C_List&, int&);
void Test_DelEl(C_List&, int&);

int main() {
    setlocale(LC_ALL, "Rus");
    int count = 0;
    C_List List;
    List.setListSize(3);
    List.CreateList();

    Test_GetProgramID(List, count);
    Test_AddEl(List, count);
    Test_DelEl(List, count);

    if (count == 3) printf("\nВсе тесты пройдены.\n");
    else printf("\nНе все тесты пройдены.\n");

    List.FreeMemory();
    if (_CrtDumpMemoryLeaks()) printf("Есть утечка памяти.");
    else printf("Утечка памяти отсутствует.");

    return 0;
}

void Test_GetProgramID(C_List list, int& count)
```

```

{
    C_Program List = list.GetProgramID(2);

    if (List.getTime() == 25326)
    {
        printf("Тест нахождения элемента по ID\t\t выполнен успешно.\n");
        count++;
    }
    else printf("Тест нахождения элемента по ID\t\t не выполнен успешно.\n");
}
void Test_AddEl(C_List& list, int& count)
{
    C_Program newProgram = list.Program2();
    int size = list.getListSize();
    list.AddEl(newProgram);

    if (list.List[size-1].getTime() != list.List[size].getTime() && list.List[size].getTime() ==
55555)
    {
        printf("Тест добавления элемента в список\t выполнен успешно.\n");
        count++;
    }
    else printf("Тест добавления элемента в список\t не выполнен успешно.\n");
}
void Test_DelEl(C_List& list, int& count)
{
    int size = list.getListSize();
    list.DeleteEl(3);
    if (size > list.getListSize())
    {
        printf("Тест функции удаления\t\t\t выполнен успешно.\n");
        count++;
    }
    else printf("Тест функции удаления\t\t\t не выполнен успешно.\n");
}

```

Main.cpp

```

#include "Program.h"
#include "List.h"

int main()
{
    setlocale(LC_ALL, "Rus");

    Menu();

    if (_CrtDumpMemoryLeaks()) printf("Есть утечка памяти.\n");
    else printf("Нет утечки памяти.\n");

    return 0;
}

void Menu()
{
    C_List List;
    C_Program getProgram;
    C_Program newProgram = List.Program2();
    int choise = 1, value = 0;
    List.setListSize(3);
    List.CreateList();
    printf("Выберите команду для работы со списком:\n");
    while (choise != 0)
    {
        printf("\n1)Вывести всё на экран\n2)Вывести 1 элемент на экран\n");
        printf("3)Добавить элемент(в конец)\n4)Удалить 1 элемент\n5)Завершение работы\n");
    }
}

```

```

printf("6)Найти программу по индексу\n=====Ваш выбор: ");
scanf("%i", &choise);

switch (choise)
{
case 1:
    List.PrintAll();
    break;
case 2:
    printf("Введите номер элемента, который надо вывести: ");
    scanf("%i", &value);
    List.PrintOneEl(value - 1);
    break;
case 3:
    List.AddEl(newProgram);
    break;
case 4:
    printf("Введите номер элемента, который хотите удалить: ");
    scanf("%i", &value);
    List.DeleteEl(value);
    break;
case 5:
    printf("Завершение работы.\n");
    choise = 0;
    break;
case 6:
    printf("Введите id элемента, которого вы хотите получить: ");
    scanf("%i", &value);
    getProgram = List.GetProgramID(value-1);
    break;
default:
    printf("Неверный символ. Повторите попытку.\n");
    break;
}
}
List.FreeMemory();
return;
}

```

List.h

```

#pragma once
#include "Program.h"
class C_List {
private:
    int listSize;

public:
    C_Program* List;
    void setListSize(int);
    int getListSize() const;
    void CreateList();
    void PrintAll() const;
    void PrintOneEl(int) const;
    void AddEl(C_Program&);
    void DeleteEl(int);
    void FreeMemory();
    C_Program GetProgramID(int) const;

    C_Program Program1();
    C_Program Program2();
};

```

List.cpp

```
#include "List.h"

void C_List::CreateList()
{
    List = new C_Program[listSize];

    for (int i = 0; i < listSize; i++)
        List[i] = Program1();
}

void C_List::setListSize(int size)
{
    listSize = size;
}

int C_List::getListSize() const
{
    return listSize;
}

void C_List::PrintAll() const
{
    printf("    Время\t\tРазмер\t\tСтроки");
    for (int i = 0; i < listSize; i++)
        PrintOneEl(i);
}

void C_List::PrintOneEl(int number) const
{
    printf("\n%2i) %-10i\t %-10i\t %-10i", number + 1, List[number].getTime(),
List[number].getSize(), List[number].getLines());
}

void C_List::AddEl(C_Program& newProgram)
{
    newProgram = Program2();
    C_Program* newList = new C_Program[listSize+1];

    for (int i = 0; i < listSize; i++)
        newList[i] = List[i];
    newList[listSize++] = newProgram;
    delete[] List;

    List = new C_Program[listSize];
    for (int i = 0; i < listSize; i++)
        List[i] = newList[i];

    delete[] newList;
    printf("Элемент добавлен.\n");
}

void C_List::DeleteEl(int index)
{
    if (listSize == 0)
    {
        printf("Список программ пуст. Возвращение с выбору действий.\n");
        return;
    }
    if (index <= 0 || index > listSize)
    {
        printf("Ошибка. Неверный номер элемента. Возвращение.\n");
        return;
    }

    C_Program* newList = new C_Program[listSize-1];

    for (int i = 0; i < index - 1; i++)
        newList[i] = List[i];
    for (int i = index - 1, j = index; i < listSize; i++, j++)
        newList[i] = List[j];
    delete[] List;
```

```

        List = new C_Program[listSize--];
        for (int i = 0; i < listSize; i++)
            List[i] = newList[i];
        delete[] newList;

        return;
    }
    void C_List::FreeMemory()
    {
        delete[] List;
    }
    C_Program C_List::GetProgramID(int id) const
    {
        C_Program temp = List[id];
        return temp;
    }

    C_Program C_List::Program1()
    {
        C_Program Program1;
        Program1.setTime(25326);
        Program1.setSize(2000);
        Program1.setLines(500);
        return Program1;
    }
    C_Program C_List::Program2()
    {
        C_Program Program2;
        Program2.setTime(55555);
        Program2.setSize(11111);
        Program2.setLines(22222);
        return Program2;
    }
}

```

Program.h

```

#include <crtdbg.h>
#define DEBUG_NEW new(_NORMAL_BLOCK, FILE, __LINE)

#include <stdio.h>
#include <locale.h>

void Menu();

class C_Program {
private:
    int TimeOfWork;        //average time of program execution
    int size;              //size of program
    int AmountOfLines;     //number of lines in code

public:
    int getTime() const;
    int getSize() const;
    int getLines() const;
    void setTime(const int);
    void setSize(const int);
    void setLines(const int);
};

```

Program.cpp

```
#include "Program.h"

int C_Program::getTime() const
{
    return TimeOfWork;
}
int C_Program::getSize() const
{
    return size;
}
int C_Program::getLines() const
{
    return AmountOfLines;
}
void C_Program::setTime(int valueTime)
{
    TimeOfWork = valueTime;
}
void C_Program::setSize(int valueSize)
{
    size = valueSize;
}
void C_Program::setLines(int valueLines)
{
    AmountOfLines = valueLines;
}
```

3. Результати роботи програми

```
=====
Ваш выбор: 1
Время      Размер      Строки
1) 25326     2000     500
2) 25326     2000     500
3) 25326     2000     500
1)Вывести всё на экран
2)Вывести 1 элемент на экран
3)Добавить элемент(в конец)
4)Удалить 1 элемент
5)Завершение работы
6)Найти программу по индексу
=====
Ваш выбор: 3
Элемент добавлен.

1)Вывести всё на экран
2)Вывести 1 элемент на экран
3)Добавить элемент(в конец)
4)Удалить 1 элемент
5)Завершение работы
6)Найти программу по индексу
=====
Ваш выбор: 1
Время      Размер      Строки
1) 25326     2000     500
2) 25326     2000     500
3) 25326     2000     500
4) 55555     11111     22222
1)Вывести всё на экран
2)Вывести 1 элемент на экран
3)Добавить элемент(в конец)
4)Удалить 1 элемент
5)Завершение работы
6)Найти программу по индексу
=====
Ваш выбор: 4
Введите номер элемента, который хотите удалить: 3

1)Вывести всё на экран
2)Вывести 1 элемент на экран
3)Добавить элемент(в конец)
4)Удалить 1 элемент
5)Завершение работы
6)Найти программу по индексу
=====
Ваш выбор: 1
Время      Размер      Строки
1) 25326     2000     500
2) 25326     2000     500
3) 55555     11111     22222
1)Вывести всё на экран
2)Вывести 1 элемент на экран
3)Добавить элемент(в конец)
4)Удалить 1 элемент
5)Завершение работы
6)Найти программу по индексу
=====
Ваш выбор: 5
Завершение работы.
Нет утечки памяти.
```

```
Тест нахождения элемента по ID      выполнен успешно.
Элемент добавлен.
Тест добавления элемента в список    выполнен успешно.
Тест функции удаления                выполнен успешно.

Все тесты пройдены.
Утечка памяти отсутствует.
```

Результат роботи тестування методів

Результат роботи програми

5. Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з класами та їх специфікаторами доступу, інкапсуляцією, константами.

Програма протестована, витоків пам'яті немає, виконується без помилок.