Автор: Татаренко А., КІТ-119а

Дата: 17 червня 2020

Лабораторна робота №5.

АГРЕГАЦІЯ ТА КОМПОЗИЦІЯ

Тема. Класи. Агрегація. Композиція. Ключові слова typedef та auto.

Мета – отримати поняття агрегація та композиція; отримати знання про призначення ключових слів typedef та auto.

1 Завдання до роботи

Індивідуальне завдання 19.

Модернізувати розроблені у попередній роботі класи таким чином:
- замінити типи даних, що використовуються при індексуванні на типи з указаної бібліотеки;
- створити власний синонім типу, визначивши його необхідність;
- створити / оновити функцію сортування масиву, де крім поля, по якому виконується сортування, передається і вказівник на функцію, яка визначає напрям сортування;
- у базовий клас додати два поля, що мають кастомний тип даних (тип даних користувача) та які будуть відображати відношення «агрегація» та «композиція», при цьому оновити методи читання та запису об’єкта;
- ввести використання ключового слова auto як специфікатор зберігання типу змінної. Визначити плюси та мінуси цього використання.

2 Розробка алгоритму розв’язання задачі.

2.1 Опис змінних

Arr stud_array; class Student; class Arr;

Класи, методи, функції, конструктори

3 Код програми

audience.h

```
#pragma once
typedef short sint;

class Aud
{
private:
	sint aud_numb;

public:
	sint get_aud_numb() const;
	void set_aud_numb(sint);

	Aud();
	Aud(sint);
	Aud(const Aud& other);
	~Aud();
};
```

basic_class.h

```
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtdbg.h>

#include <iostream>
#include <fstream>
#include <sstream>
#include <regex>

#include "audience.h"
#include "faculty.h"

#define El 3

using std::cout;
using std::cin;
using std::endl;
using std::stringstream;
using std::ofstream;
using std::ifstream;
using std::regex;

typedef bool (comp)(const int&, const int&);

class Student
{
private:

	int age;
	int number_stud;
	int middle_mark;
	string name;
	bool debt;
	int prog_d;
	Aud audience;
	Fac faculty;

public:
```

```
        Student();

        Student(int a, int n, int m, string na, bool d, int pd, sint ad, string fc);

        Student(const Student &other);

        int get_number_stud() const;

        int get_age() const;

        int get_middle_mark() const;

        string get_name() const;

        bool get_debt() const;

        int get_prog_d() const;

        sint get_aud();

        string get_fac();

        ~Student()
        {
                cout << "Вызвался деструктор!" << endl;
        }
};
```

class_list.h:

```
#pragma once
#include "basic_class.h"

class Arr
{
private:

        Student* array_stud;

        int count = El;

public:

        void create_array();

        void print_array(int) const;

        void delete_one();

        void add();

        string select() const;

        void fill_array();

        int count_plus();

        int count_minus();

        void print_one(Student) const;

        void delete_array();
```

```
        Student Construct(int);

        int prog_d_rand(bool);

        void find_debt();

        void in_f();

        void from_f(int);

        int get_count() const;

        int str_in_file(string) const;

        void set_count(int);

        void regex_task();

        void sort(comp);

        static bool sort_forward(const int&, const int&);

        static bool sort_back(const int&, const int&);

};
```

faculty.h

```
#pragma once

#include <string>
using std::string;

class Fac
{
private:
        string fname;
public:
        string get_fname() const;
        void set_fname(string);

        Fac();
        Fac(string);
        Fac(const Fac& other);
        ~Fac();
};
```

menu.h:

```
#pragma once
#include "class_list.h"


void menu(Arr);
```

test.h:

```
#pragma once
#include "class_list.h"

bool test_count_plus(Arr);
```

```
bool test_count_minus(Arr);
```

audience.cpp

```
#include "audience.h"

sint Aud::get_aud_numb() const
{
	return aud_numb;
}

void Aud::set_aud_numb(sint a_n)
{
	aud_numb = a_n;
}

Aud::Aud() : aud_numb(0)
{
}

Aud::Aud(sint aud_numb) : aud_numb(aud_numb)
{
}

Aud::Aud(const Aud& other) : aud_numb(other.aud_numb)
{
}

Aud::~Aud()
{
}
```

basic_class.cpp:

```
#include "basic_class.h"

Student::Student() : age(0), number_stud(0), middle_mark(0), name("Name"), debt(0),
prog_d(0), audience(0), faculty("Non")
{
	cout << "Вызван стандартый конструктор!" << endl;
}

Student::Student(int a, int n, int m, string na, bool d, int pd, sint an, string fc) :
age(a), number_stud(n), middle_mark(m), name(na), debt(d), prog_d(pd), audience(an),
faculty(fc)
{
	cout << "Вызван конструктор с параметрами!" << endl;
}

Student::Student(const Student &other) : age(other.age), number_stud(other.number_stud),
middle_mark(other.middle_mark), name(other.name), debt(other.debt), prog_d(other.prog_d),
audience(other.audience), faculty(other.faculty)
{
	cout << "Вызван конструктор копирования!" << endl;
}

int Student::get_number_stud() const
{
	return number_stud;
}
```

```
int Student::get_age() const
{
	return age;
}

int Student::get_middle_mark() const
{
	return middle_mark;
}


string Student::get_name() const
{
	return name;
}

bool Student::get_debt() const
{
	return debt;
}

int Student::get_prog_d() const
{
	return prog_d;
}

sint Student::get_aud()
{
	return audience.get_aud_numb();
}

string Student::get_fac()
{
	return faculty.get_fname();
}
```

class_list.cpp:

```
#include "class_list.h"

void Arr::create_array()
{
	int c = get_count();

	array_stud = new Student[c];
}

void Arr::fill_array()
{
	int c = get_count();

	for (size_t i = 0; i < c; i++)
	{
		array_stud[i] = Construct(i);
	}

}

Student Arr::Construct(int i)
{
	bool d = 0;

	if (i == 0)
	{
```

```
			d = rand() % 2;
			Student st0(i + 17, i + 1, i + 10 / 2, "Dani", d, prog_d_rand(d), 15,
"KIT");
			return st0;
		}
		else if (i == 1)
		{
			d = rand() % 2;
			Student st1(i + 17, i + 1, i + 10 / 2, "Peter", d, prog_d_rand(d), 15,
"KIT");
			return st1;
		}
		else if (i == 2)
		{
			d = rand() % 2;
			Student st2(i + 17, i + 1, i + 10 / 2, "Donald Tramp", d, prog_d_rand(d),
36, "PIT");
			return st2;
		}
		else if (i == 3)
		{
			d = rand() % 2;
			Student st2(i + 17, i + 1, i + 10 / 2, "Vladimir", d, prog_d_rand(d), 87,
"PIT");
			return st2;
		}

		d = rand() % 2;
		Student st(i + 17, i + 1, i + 10 / 2, "Edgar", d, prog_d_rand(d), 17, "GM");
		return st;
}

void Arr::print_array(int c) const
{
		for (size_t i = 0; i < 50; i++)
			cout << "-";
		cout << endl;
		////////////////
		for (size_t i = 0; i < c; i++)
		{
			cout << "Номер: " << array_stud[i].get_number_stud() << "\tВозраст: " <<
array_stud[i].get_age() << "\t\tСредний балл: " << array_stud[i].get_middle_mark() <<
"\t\tИмя: " << array_stud[i].get_name() << "\t\tАудитория: " << array_stud[i].get_aud()
<< "\t\tФакультет: " << array_stud[i].get_fac() << endl;
		}
		////////////////
		for (size_t i = 0; i < 50; i++)
			cout << "-";
		cout << endl;
}

void Arr::delete_one()
{
		int input, i_arr = 0;
		cout << "Введите номер: ";
		cin >> input;

		for (size_t i = 0; i < El; i++)
			if (input == array_stud[i].get_number_stud())
			{
				count_minus();
				int c = get_count();

				Student* array_stud_new = new Student[c];
```

```
                        for (size_t i = 0; i < c + 1; i++)
                        {
                                if (input == array_stud[i].get_number_stud())
                                        continue;
                                array_stud_new[i_arr++] = array_stud[i];
                        }

                        delete[] array_stud;

                        array_stud = new Student[c];

                        for (size_t i = 0; i < c; i++)
                        {
                                array_stud[i] = array_stud_new[i];
                        }

                        delete[] array_stud_new;

                        print_array(c);

                        return;
                }
        ////////////////
        for (size_t i = 0; i < 50; i++)
                cout << "-";
        cout << endl;
        ////////////////
        cout << "Этого студента не существует!" << endl;
        ////////////////
        for (size_t i = 0; i < 50; i++)
                cout << "-";
        cout << endl;
        ////////////////
}

void Arr::add()
{
        count_plus();
        int c = get_count();

        Student* array_stud_new = new Student[c];

        size_t i = 0;

        for (i; i < c - 1; i++)
        {
                array_stud_new[i] = array_stud[i];
        }

        array_stud_new[c - 1] = Construct(i);

        delete[] array_stud;

        array_stud = new Student[c];

        for (i = 0; i < c; i++)
        {
                array_stud[i] = array_stud_new[i];
        }

        delete[] array_stud_new;

        print_array(c);
}
```

```
int Arr::count_plus()
{
	count = count + 1;

	return count;
}

int Arr::count_minus()
{
	count = count - 1;

	return count;
}

string Arr::select() const
{
	int input = 0;
	cout << "Введите номер: ";
	cin >> input;
	int c = get_count();

	for (size_t i = 0; i < c; i++)
	{
		if (input == array_stud[i].get_number_stud())
		{
			string info;
			stringstream ss;
			ss << "Номер: " << array_stud[i].get_number_stud() << "\tВозраст: "
<< array_stud[i].get_age() << "\t\tСредний балл: " << array_stud[i].get_middle_mark() <<
"\t\tИмя: " << array_stud[i].get_name() << endl;
			info = ss.str();

			return info;
		}
	}
	////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////
	cout << "Этого студента не существует!" << endl;
	////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////
}

void Arr::print_one(Student stud) const
{
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////////////
	cout << "Номер: " << stud.get_number_stud() << "\tВозраст: " << stud.get_age() <<
"\t\tСредний балл: " << stud.get_middle_mark() << "\t\tИмя: " << stud.get_name() <<
"\t\tАудитория: " << stud.get_aud() << "\t\tФакультет: " << stud.get_fac() << endl;
	////////////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
}

void Arr::delete_array()
{
```

```
	delete[] array_stud;
}

int Arr::prog_d_rand(bool d)
{
	int pd = 0;

	if (d == 1)
		pd = rand() % 50;

	return pd;
}

void Arr::find_debt()
{
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////////////
	for(size_t i = 0; i < count; i++)
		if (array_stud[i].get_debt() == 1)
		{
			cout << "Номер: " << array_stud[i].get_number_stud() << "\tВозраст:
" << array_stud[i].get_age() << "\t\tСредний балл: " << array_stud[i].get_middle_mark()
<< "\t\tИмя: " << array_stud[i].get_name() << "\t\tДолг: Есть" << "\t\tДолг по
программированию: " << array_stud[i].get_prog_d() << "%" << endl;
		}
	////////////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;

}

void Arr::in_f()
{
	ofstream file;
	file.open("of.txt", ofstream::app);

	if (!file.is_open())
	{
		cout << "Файл не открыт!" << endl;
		return;
	}

	string info;
	stringstream ss;
	int c = get_count();

	for (size_t i = 0; i < c; i++)
	{
		ss << "Номер: " << array_stud[i].get_number_stud() << "\tВозраст: " <<
array_stud[i].get_age() << "\t\tСредний балл: " << array_stud[i].get_middle_mark() <<
"\t\tИмя: " << array_stud[i].get_name() << "\t\tАудитория: " << array_stud[i].get_aud()
<< "\t\tФакультет: " << array_stud[i].get_fac() << endl;
	}

	for (size_t i = 0; i < 80; i++)
		ss << "-";
	ss << endl;

	info = ss.str();
	file << info;

	file.close();
```

```
	////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////
	cout << "Файл записан успешно!" << endl;
	////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////

}

void Arr::from_f(int size)
{
	string s;
	regex varEn("([\\d]* [\\d]* [\\d]* [A-Z]+[\\w,.;:-]* [0|1] [\\d]* [\\d]* [A-
Z]*)");

	ifstream file;
	file.open("if.txt");

	if (!file.is_open())
	{
		cout << "Файл не открыт!" << endl;
		return;
	}

	delete[] array_stud;
	array_stud = new Student[size];

	for (size_t i = 0; i < size; i++)
	{
		getline(file, s);
		if (regex_match(s, varEn))
		{
			std::istringstream iss(s);
			int age;
			int number;
			int middle_mark;
			string name;
			bool debt;
			int prog_d;
			sint audience;
			string faculty;

			iss >> age;
			iss >> number;
			iss >> middle_mark;
			iss >> name;
			iss >> debt;
			iss >> prog_d;
			iss >> audience;
			iss >> faculty;

			Student file_el(age, number, middle_mark, name, debt, prog_d,
audience, faculty);
			array_stud[i] = file_el;
		}
	}

	file.close();
```

```
	cout << "Чтение с файла успешно!" << endl << endl;

	set_count(size);

	print_array(size);

}

int Arr::get_count() const
{
	return count;

}

int Arr::str_in_file(string fileName) const
{
	int size = 0;
	int c = get_count();
	string line;
	ifstream file(fileName);

	if (!file.is_open() || c == 0)
	{
		cout << "There is no such file" << endl << endl;
		return 0;
	}

	while (getline(file, line))
	{
		size++;
	}
	file.close();

	return size;
}

void Arr::set_count(int c)
{
	count = c;

}

void Arr::regex_task()
{
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
	////////////////
	regex regular("(^[A-ZÀ-ß]+[\\wÀ-ßà-ÿ,.;:-]* [\\wÀ-ßà-ÿ,.;:-]+)");
	int listSize = get_count();

	for (size_t i = 0; i < listSize; i++)
	{
		if (regex_match(array_stud[i].get_name(), regular))
			print_one(array_stud[i]);

	}

	cout << endl;
	////////////////
	for (size_t i = 0; i < 50; i++)
		cout << "-";
	cout << endl;
}
```

```
bool Arr::sort_forward(const int& a, const int& b)
{
	return a > b;
}

bool Arr::sort_back(const int& a, const int& b)
{
	return a < b;
}

void Arr::sort(comp condition)
{
	Student temp;
	int size = get_count();
	bool pr;

		do {
			pr = 0;
			for (size_t i = 0; i < size - 1; i++)
			{
				if (condition(array_stud[i].get_number_stud(), array_stud[i +
1].get_number_stud()))
				{
					temp = array_stud[i];
					array_stud[i] = array_stud[i + 1];
					array_stud[i + 1] = temp;
					pr = 1;
				}
			}
		} while (pr == 1);

	print_array(size);
}
```

main.cpp:

```
/**
 * @mainpage
 * <b> Лабораторна робота № 5. <br/> АГРЕГАЦІЯ ТА КОМПОЗИЦІЯ </b>
 * <br/><b><i>Мета роботи:</i></b>:  отримати знання про базові регулярні вирази та
досвід роботи із застосування їх на практиці. <br/>
 * <b><i>Індивідуальне завдання 19:</i></b>
 *     Поширити попередню лабораторну роботу. <br/>
 * <br/>
 * <br/> <b> Arr p; class Student; class Arr; </b> Класи, методи, функції, конструктори,
потоки </b>
 * @author Tatarenko A.
 * @date 29-may-2020
 * @version 1.0
 */

#include "menu.h"

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

	Arr stud_array;

	auto input = 0;

	cout << "Введите 1 чтобы создать массив, 2 чтобы выйти: ";
	cin >> input;
```

```
	if (input == 1)
	{
		stud_array.create_array();
		cout << "Список созданый с помощью конструктора:" << endl;
		stud_array.print_array(stud_array.get_count());

		cout << "Рандомное заполнение списка:" << endl;
		stud_array.fill_array();
		stud_array.print_array(stud_array.get_count());

		menu(stud_array);

		stud_array.delete_array();

	}

	////////////////////////////////////////////////////////
	int l = _CrtDumpMemoryLeaks(); // Контроль витоку пам'яті
	if (l == 0)
		cout << "Утечки памяти не обнаруженно!" << endl;
	else
		cout << "Обнаружена утечка памяти!" << endl;

	system("PAUSE");
	return 0;
}
```

faculty.cpp

```
#include "faculty.h"

string Fac::get_fname() const
{
	return fname;
}

void Fac::set_fname(string fn)
{
	fname = fn;
}

Fac::Fac() : fname("Non")
{
}

Fac::Fac(string fn) : fname(fn)
{
}

Fac::Fac(const Fac& other) : fname(other.fname)
{
}

Fac::~Fac()
{
}
```

menu.cpp:

```
#include "menu.h"

void menu(Arr stud_array)
```

```
{
	int input = 0;

	cout << "1) Добавить объект в конец;" << endl << "2) Удалить объект;" << endl << "3) Выбрать объект;" << endl << "4) Вывести все объекты;" << endl << "5) Поиск должников;" << endl << "6) Запись в файл;" << endl << "7) Читать с файла;" << endl << "8) REGEX TASK;" << endl << "9) Сортировать;" << endl << "10) Выйти;" << endl << "Ваше действие: ";
	cin >> input;

	switch (input)
	{
	case 1:
		stud_array.add();
		break;
	case 2:
		stud_array.delete_one();
		break;
	case 3:
		cout << stud_array.select();
		break;
	case 4:
		stud_array.print_array(stud_array.get_count());
		break;
	case 5:
		stud_array.find_debt();
		break;
	case 6:
		stud_array.in_f();
		break;
	case 7:
		stud_array.from_f(stud_array.str_in_file("if.txt"));
		break;
	case 8:
		stud_array.regex_task();
		break;
	case 9:
		cout << "Выберете тип сортировки:" << endl << "Сортировка по возрастанию;" << endl << "Сортировка по убыванию;" << endl << endl;
		int c;
		cin >> c;

		if (c == 1)
			stud_array.sort(stud_array.sort_forward);
		else if (c == 2)
			stud_array.sort(stud_array.sort_back);
		else
		{
			cout << "НЕВЕРНОЕ ДЕЙСТВИЕ!" << endl;
			////////////////
			for (size_t i = 0; i < 50; i++)
				cout << "-";
			cout << endl;
		}
		break;
	case 10:
		return;
	default:
		break;
	}

	menu(stud_array);
}
```

test.cpp:

```
#include "test.h"

int main()
{
	Arr stud_array;

	if (test_count_plus(stud_array))
		cout << "Function count_plus is right!" << endl;
	else
		cout << "Function count_plus is not right!" << endl;

	if (test_count_minus(stud_array))
		cout << "Function count_minus is right!" << endl;
	else
		cout << "Function count_minus is not right!" << endl;

	system("PAUSE");
}

bool test_count_plus(Arr p)
{
	int exp = El + 1;
	int cnt = p.count_plus();

	if (cnt == exp)
		return 1;
	else
		return 0;
}

bool test_count_minus(Arr p)
{
	int exp = El - 1;
	int cnt = p.count_minus();

	if (cnt == exp)
		return 1;
	else
		return 0;

}
```

4 Результати тестування

```
Введите 1 чтобы создать массив, 2 чтобы выйти: 1
Вызван стандартый конструктор!
Вызван стандартый конструктор!
Вызван стандартый конструктор!
Список созданый с помощью конструктора:
-------------------------------------------------
Номер: 0        Возраст: 0              Средний балл: 0         Имя: Name               Аудитория: 0            Факультет: Non
Номер: 0        Возраст: 0              Средний балл: 0         Имя: Name               Аудитория: 0            Факультет: Non
Номер: 0        Возраст: 0              Средний балл: 0         Имя: Name               Аудитория: 0            Факультет: Non
-------------------------------------------------
Рандомное заполнение списка:
Вызван конструктор с параметрами!
Вызван конструктор копирования!
Вызвался деструктор!
Вызвался деструктор!
Вызван конструктор с параметрами!
Вызван конструктор копирования!
Вызвался деструктор!
Вызвался деструктор!
Вызван конструктор с параметрами!
Вызван конструктор копирования!
Вызвался деструктор!
Вызвался деструктор!
-------------------------------------------------
Номер: 1        Возраст: 17             Средний балл: 5         Имя: Dani               Аудитория: 15           Факультет: KIT
Номер: 2        Возраст: 18             Средний балл: 6         Имя: Peter              Аудитория: 15           Факультет: KIT
Номер: 3        Возраст: 19             Средний балл: 7         Имя: Donald Tramp               Аудитория: 36
Факультет: PIT
-------------------------------------------------
1) Добавить объект в конец;
2) Удалить объект;
3) Выбрать объект;
4) Вывести все объекты;
5) Поиск должников;
6) Запись в файл;
7) Читать с файла;
8) REGEX TASK;
9) Сортировать;
10) Выйти;
Ваше действие:
```

5 Опис результатів

Програма виводить список студентів. Виконує видалення, додавання, вибір об’єкта, виводить інформацію про заборгованість. А також виконує запис і читання з файлу, та перевіряє читання через базові регулярні вирази.

Висновок:

Протягом цієї роботи було отримано базові знання про поняття агрегація та композиція, та про призначення ключових слів typedef та auto.