Група: ПП-16

Студент: Якіб'юк Ігор

Варіант №: 14

Дата захисту:

Кафедра: САП

Дисципліна: Об'єктно-орієнтоване програмування

Перевірила (в): Чумакевич В. В.

ЗВІТ

з лабораторної роботи №2

на тему "Класи та об'єкти. Композиція класів та об'єктів інших класів"

Мета роботи: Навчитись розробляти класи для опису об'єктів заданої предметної області, створювати об'єкти та працювати з ними. Ознайомитись з поняттям композиції, навчитись розробляти класи, які включають об'єкти інших класів.

Індивідуальне завдання:

1. Реалізувати клас, що містить такі методи:

   - конструктор за замовчуванням;

   - конструктор з параметрами;

   - функції для зміни кожного з полів класу;

   - функції для виведення кожного з полів класу.

   Реалізувати клас Money, котрий містить інформацію про: номінал, рік випуску куп'юри, курс валюти відносно української гривні.

2. Створити клас, що міститиме масив об'єктів попереднього класу. Клас повинен містити наступні поля: статичний масив об'єктів, реальна кількість елементів в масиві, максимальна кількість елементів в масиві. А також наступні функції:
   - додавання нового елемента в масив,
   - редагування інформації про об'єкт масиву з вказаним номером,
   - видалення об'єкта з масиву,
   - виведення на екран інформації про об'єкт масиву з вказаним номером,
   - виведення на екран всіх елементів масиву

Реалізувати клас Money, котрий містить інформацію про: номінал, рік випуску куп'юри, курс валюти відносно української гривні.

Додатково реалізувати:

- метод сортування куп'юри за номіналом,

- метод для виведення на екран списку куп'юр 2018 року випуску.

Етапи розв’язку:

Для виконання цих завдань я використав базові бібліотеки такі, як: iostream, windows.h. А також algorithm, string. Для зручності і модульності коду розпорошив його по файлах (умовно кожен клас в окремому файлі з декларацією полів і подальшою їх реалізацією). Для використання їх в main використовую #include.

Програмна реалізація з коментярями:

File Lab2.cpp

```
#include "Money.h"
#include "ListMoney.h"

Money::Money(string nominal, int year, double rate, int id) {
	this->nominalValue = nominal;
	this->releaseYear = year;
	this->exchangeRate = rate;
	this->id = id;
}

void Money::changeData(string nominal, int year, double rate) {
	this->nominalValue = nominal.empty() ? this->nominalValue : nominal;
	this->releaseYear = year ? year : this->releaseYear;
	this->exchangeRate = rate ? rate : this->exchangeRate;
}

int Money::getId()  {
	return this->id;
}

int Money::getYear() {
	return this->releaseYear;
}

string Money::getNominal() {
	return this->nominalValue;
}

void Money::getData() {
	cout << "Nominal value: " << this->nominalValue << "\t" << "release year: " <<
this->releaseYear << "\t" << "exchange rate: " << this->exchangeRate << endl;
}
```

File Money.h

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

using namespace std;

class Money {
private:
	string nominalValue;
	int releaseYear;
	double exchangeRate;
	int id;

public:
```

```
        Money(string nominal, int year, double rate, int id);

        void changeData(string nominal, int year, double rate);

        int getId();

        int getYear();

        string getNominal();

        void getData();

};
```

File Money.cpp

```
#include "Money.h"
#include "ListMoney.h"

Money::Money(string nominal, int year, double rate, int id) {
        this->nominalValue = nominal;
        this->releaseYear = year;
        this->exchangeRate = rate;
        this->id = id;
}

void Money::changeData(string nominal, int year, double rate) {
        this->nominalValue = nominal.empty() ? this->nominalValue : nominal;
        this->releaseYear = year ? year : this->releaseYear;
        this->exchangeRate = rate ? rate : this->exchangeRate;
}

int Money::getId()  {
        return this->id;
}

int Money::getYear() {
        return this->releaseYear;
}

string Money::getNominal() {
        return this->nominalValue;
}

void Money::getData() {
        cout << "Nominal value: " << this->nominalValue << "\t" << "release year: " <<
this->releaseYear << "\t" << "exchange rate: " << this->exchangeRate << endl;
}
```

File ListMoney.h

```
#pragma once
#include <string>
#include <iostream>
#include <vector>
#include "Money.h"

class ListMoney {
private:
        vector<Money> listMoneys;
        int listLength = listMoneys.size();
        int maxListLength = 10;

public:
```

```
	ListMoney();

	void changeMaxLength(int limit);

		void addNewObj(Money obj);

		void findElement(int id);

		void findChangeElement(int id, string nominal, int year, double rate);

		void getAllList();

		void deleteMoney(int id);

		void get2018Elements(int year);

		void sortCurrencyByNominal();
};
```

File ListMoney.cpp

```
#include "ListMoney.h"
#include "Money.h"
#include <algorithm>

ListMoney::ListMoney() {
}

void ListMoney::changeMaxLength(int limit) {
	this->maxListLength = limit;
}

void ListMoney::addNewObj(Money obj) {
	if (this->listLength < this->maxListLength) {
		this->listMoneys.push_back(obj);
	}
	else {
		cout << "overfilled";
	}
}

void ListMoney::findElement(int id) {
	for (Money& m : this->listMoneys) {
		if (m.getId() == id) {
			m.getData();
		}
	}
}

void ListMoney::findChangeElement(int id, string nominal, int year, double rate) {
	for (Money& m : this->listMoneys) {
		if (m.getId() == id) {
			m.changeData(nominal, year, rate);
			m.getData();
		}
	}
}

void ListMoney::getAllList() {
	for (Money& m : this->listMoneys) {
		m.getData();
	}
}
```

```
void ListMoney::deleteMoney(int id) {
	this->listMoneys.erase(remove_if(this->listMoneys.begin(), this-
>listMoneys.end(), [id](Money m) { return m.getId() == id; }), this-
>listMoneys.end());
}

void ListMoney::get2018Elements(int year) {
	for (Money& m : this->listMoneys) {
		if (m.getYear() == year) {
			m.getData();
		}
	}
}

void ListMoney::sortCurrencyByNominal() {
	sort(this->listMoneys.begin(), this->listMoneys.end(),
		[](Money a, Money b) { return a.getNominal() < b.getNominal(); });
	this->getAllList();
}
```

Результат виконання програми:

Nominal value: dollar release year: 1792 exchange rate: 38.1

Nominal value: euro release year: 1999 exchange rate: 39.6

Nominal value: pound release year: 1124 exchange rate: 43.2

Nominal value: koruna ?esk? release year: 1993 exchange rate: 1.62

Nominal value: Australian dollar release year: 1966 exchange rate: 25.28

Nominal value: euro release year: 1996 exchange rate: 1

Nominal value: dollar2 release year: 12 exchange rate: 12

Nominal value: dollar2 release year: 12 exchange rate: 12

Nominal value: euro release year: 1999 exchange rate: 39.6

Nominal value: pound release year: 1124 exchange rate: 43.2

Nominal value: koruna ?esk? release year: 1993 exchange rate: 1.62

Nominal value: Australian dollar release year: 1966 exchange rate: 25.28

Nominal value: euro release year: 1996 exchange rate: 1

D:\OOP\Labs\Lab2\x64\Debug\Lab2.exe (process 16764) exited with code 0.

Press any key to close this window . . .

Висновок:

На даній лабораторній роботі я навчився створювати класи, а пізніше об’єкти на їх основі, працювати з ними. Також Ознайомився з модифікаторами доступу і основними функціями (методами класу): конструктор, деструктор. Закріпив отримані знання на практиці, виконавши індивідуальні завдання (створення класів(об’єктів) і робота з ними).

Відповіді на контрольні запитання:

1. **Що таке клас?**

   Деякий проєкт, з якого створюють об’єкти. Він містить властивості і методи.

2. **Що таке об’єкт?**

   Об’єкт – сутність, екземпляр класу, що має власну поведінку, описану у класі.

3. **Для чого потрібен конструктор класу?**

   Встановити початковий стан об’єкту шляхом ініціалізації атрибутів.

4. **Для чого потрібен деструктор класу?**

   Для виконання операції деініціалізації (видалення) об’єкту, звільнення пам’яті.

5. **Рівні доступу до полів та методів класу?**

   Public
   Private
   Protected

6. **Що таке композиція? Для чого її використовують?**

   Ц е створення об'єктів існуючих класів як елементів інших класів