

Проектування і реалізація програми з масивами об'єктів, вказівниками та посиланнями на них

Мета роботи — засвоєння поняття масиву об'єктів, вказівників та посилань на об'єкти класів; набуття практичних навичок їх оголошення та використання.

Основні завдання роботи

Розробити і реалізувати програмний додаток нарахування заробітної плати співробітникам фірми. Для збереження даних про співробітників використати масиви об'єктів, а для доступу до функцій — вказівники. Для цього:

1. Розробити і реалізувати програмно клас “Особа”.
2. Розробити і реалізувати програмно функції, які враховують такі вимоги:
 - 2.1. Кожному співробітнику заробітна плата нараховується, виходячи з тарифної сітки, у якій закладено такі позиції: “Спеціаліст”, “Спеціаліст 1 категорії”, “Спеціаліст 2 категорії”, “Спеціаліст вищої категорії”, “Начальник відділу”, “Директор”.
 - 2.2. Кожному співробітнику призначається додатково до ставки премія за принципом: якщо заробітна плата менша від середньої заробітної плати усіх співробітників, то розмір премії становить 50% від заробітної плати, в інших випадках — 30% від заробітної плати.
 - 2.3. Реалізувати програмно функцію, яка визначає суму, що видається на руки (заробітна плата плюс премія).

3. Продемонструвати роботу створеного класу і його функцій в основній частині програми.

Основні теоретичні відомості

1. Масиви об'єктів

Об'єкти — це змінні, які мають такі самі можливості, що й змінні інших типів. Тому допустимим є вкладення об'єктів у масив. Синтаксис оголошення масиву об'єктів аналогічний тому, який використовується для оголошення масиву змінних будь-якого іншого типу. Доступ до масивів об'єктів аналогічний доступу до масивів змінних будь-якого іншого типу.

Масиви об'єктів створюються таким чином:

```
ім'я_класу ім'я_масиву [кількість_елементів_масиву];
```

Якщо клас містить конструктор, масив об'єктів може бути ініціалізований. Наприклад, для об'єкта *ob* класу *myclass* (конструктор якого ініціалізує лише одну змінну типу *int*) масив об'єктів можна ініціалізувати так:

```
myclass ob[4] = { -1, -2, -3, -4 };  
//тобто присвоюємо ob[0].x=-1; ...; ob[3].x=-4
```

Якщо ж конструктор ініціалізує більше однієї змінної, тоді під час ініціалізації масиву об'єктів слід передавати об'єкти класу. Наприклад:

```
class myclass {  
    int x,y;  
public:  
    myclass(int a, int b) {x = a; y= b;}; //конструктор  
};  
...  
myclass ob[2] = { myclass(10, 2), myclass(-3, -4) };  
//масив об'єктів, присвоюємо значення  
//ob[0].x=10; ob[0].y=2; ob[1].x=-3; ob[1].y=-4;
```

Можна також оголосити двовимірний масив об'єктів.

Наприклад, щоб оголосити масив об'єктів попереднього класу розмірності 4×2, необхідно написати у програмі таке:

```
myclass obj[4][2] = { myclass (1, 2), myclass (3, 4),  
                      myclass (5, 6), myclass (7, 8),  
                      myclass (9, 10), myclass (11, 12),  
                      myclass (13, 14), myclass (15, 16)};
```

2. Вказівники на об'єкти

Доступ до об'єкта можна отримати також через вказівник на цей об'єкт. Для цього використовують оператор “стрілка” (->). Вказівник на об'єкт оголошується так само, як і вказівник на змінну будь-якого іншого типу. Для цього треба задати ім'я класу цього об'єкта, а потім ім'я змінної із зірочкою перед ним.

Для отримання адреси об'єкта перед його іменем (або ідентифікатором) потрібно поставити оператор “&” так само, як це робиться для отримання адреси змінної іншого типу.

Арифметика вказівників на об'єкт аналогічна арифметиці вказівників на дані будь-якого іншого типу: її використовує стосовно об'єкта. Як і вказівник будь-якого типу, під час інкрементації на об'єкт вказівник буде вказувати на наступний об'єкт такого самого типу. Якщо вказівник на об'єкт декрементується, то він починає вказувати на попередній об'єкт.

Приклад. Вказівники на об'єкти класу:

```
#include <iostream>  
class myclass { int a;  
public:  
    myclass(int x); // конструктор  
    int get();  
};  
myclass::myclass(int x) { a = x; }  
int myclass::get() { return a; }  
int main() {  
    myclass ob(120); // оголошення об'єкта  
    myclass *p; // оголошення вказівника на об'єкт  
    p = &ob; // запам'ятовування адреси ob у p  
    cout << "Доступ через об'єкт: " << ob.get();  
    cout << "\n";  
}
```

```
cout << "Доступ через вказівник: " << p->get();  
return 0;  
}
```

3. Вказівник *this*

У C++ є спеціальний вказівник *this*. Це — вказівник, який автоматично передається будь-якій функції-члену класу при її виклику і вказує на об'єкт, що генерує цей виклик. Наприклад:

```
ob.f(); // нехай ob - це об'єкт
```

Функції *f()* автоматично передається вказівник на об'єкт *ob*. Цей вказівник називається *this*. Вказівник *this* передається лише функціям-членам класу.

Порядок виконання роботи

1. Для реалізації програмного коду оголосити та визначити клас *особа*. У цьому класі слід оголосити закриті змінні, що зберігають такі дані: прізвище та ініціали співробітника, заробітну плату, премію, ставку, позицію з тарифної сітки.

2. Додати у клас відкриті конструктори. Один з конструкторів оголосити без параметрів, інший з параметрами для ініціалізації об'єктів класу. Конструктор не повинен ініціалізувати значення ставки, натомість він має викликати функцію, яка її визначає.

3. Додати у клас деструктор.

4. Додати у клас функцію визначення ставки відповідно до позиції у тарифній сітці.

5. Додати у клас функцію призначення премій. Врахувати, що премія призначається за принципом: якщо заробітна плата менша, ніж середня зарплата усіх співробітників, то розмір премії становить 50% від заробітної плати, в інших випадках — 30% від заробітної плати.

6. Додати функцію визначення заробітної плати.

9. Додати функцію виведення даних про співробітника.

7. В основній частині програми створити масив об'єктів. Доступ до елементів масиву показати через вказівник.

8. Програма повинна містити меню, яке дозволить перевірити основні функції, створені у програмі.

Контрольні завдання та запитання

1. Як оголосити вказівник на об'єкт?
2. Що таке вказівник *this*?
3. Як звернутися у програмі до елементів масиву об'єктів?
4. Як отримати доступ елементу масива об'єктів до відкритих членів класу?
5. Якими способами можна виконати ініціалізацію масиву об'єктів?
6. Як оголосити двовимірний масив об'єктів?
7. Дано прототип конструктора:
`myclass(int, char, char*);`
Оголосіть масив із чотирьох об'єктів.