**Лекція №10. Синтаксис оголошення класу. Специфікатори доступу.**

Клас являє собою абстрактний тип даних, що визначається користувачем і являє собою модель реального об’єкта у вигляді даних та функцій для роботи з ними.

Б’єрн Страуструп:

"В языке С++ есть два вида типов, определенных пользователем: классы и перечисления. Классы носят намного более общий характер и играют более важную роль в программировании, поэтому мы сосредоточим свое внимание в первую очередь на них. Класс непосредственно выражает некое понятие в программе. *Класс* (class) — это тип, определенный пользователем. Он определяет, как представляются объекты этого класса, как они создаются, используются и уничтожаются. Если вы размышляете о чем-то как об отдельной сущности, то, вполне возможно, должны определить класс, представляющий эту “вещь” в вашей программе. Примерами являются вектор, матрица, поток ввода, строка, быстрое преобразование Фурье, клапанный регулятор, рука робота, драйвер устройства, рисунок на экране, диалоговое окно, график, окно, термометр и часы.

…

Класс — это тип, определенный пользователем. Он состоит из встроенных типов, других типов, определенных пользователем, и функций. Компоненты, использованные при определении класса, называются его *членами* (members). Класс может содержать несколько членов, а может и не иметь ни одного члена."

Оголошення класу має таку форму:

class <сім’я класу> : <список класів-батьків>

{ public:             // доступно для всіх **відкритий доступ**

<дані, методи, властивості, події>

protected: // доступно тільки нащадка **захищений доступ**

<дані, методи, властивості, події>

private:   // доступно тільки в класі **закритий доступ**

<дані, методи, властивості, події>

} <список змінних>

Визначення або оголошення класу починається з ключового слова class. Далі у фігурних дужках ({}) описується тіло класу. Визначення класу закінчується крапкою з комою. Тіло класу включає дані (члени-дані), функції (члени-функції) і специфікатори доступу до членів класу. Специфікатори доступу до членів класу завжди закінчуються двокрапкою (:) і можуть з'являтися у визначенні класу багато разів і у будь-якому порядку.

Властивість ООП - можливість успадкування даних і функцій. У випадку, коли один клас успадковує інший, базовий клас називають **батьківським**, а той, що успадковує, — похідним або **нащадком**.

Розміщення описів змінних і функції з їх обробки, тобто полів та методів в одному класі називається **інкапсуляцією**.

Класи в С++ мають три різних рівні доступу до своїх елементів, тобто даних (полів) і функцій (методів):

* закриті елементи (private);
* захищені елементи (protected);
* відкриті елементи (public).

До даних у закритому розділі (private) мають доступ тільки функції-елементи свого класу. Класам-нащадкам забороняється доступ до закритих даних своїх базових (батьківських) класів. За замовчуванням усі дані-члени класу мають атрибут private(закритий).

До даних у захищеному розділі (protected) мають доступ функції свого класу і функції класів-нащадків.

У свою чергу до даних відкритого розділу (public) можуть звертатися будь-які функції.

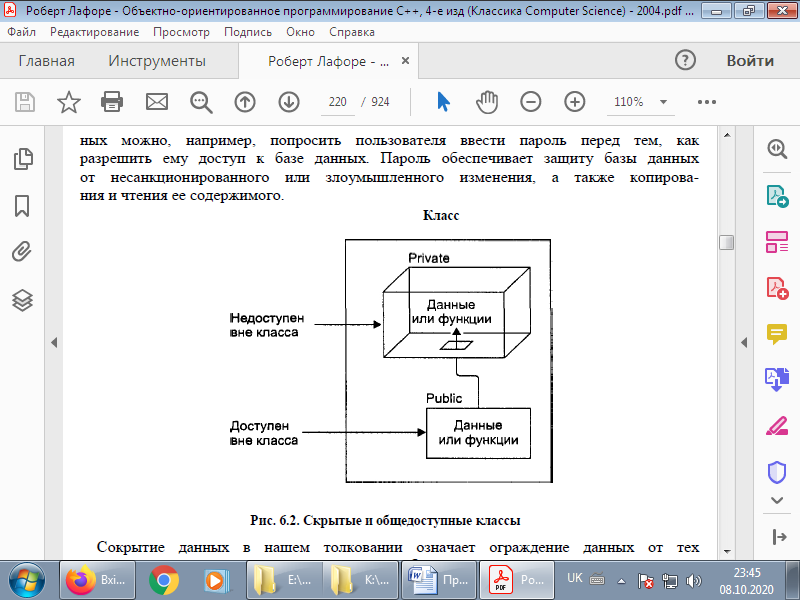


Рис.1 Приховані та загальнодоступні класи

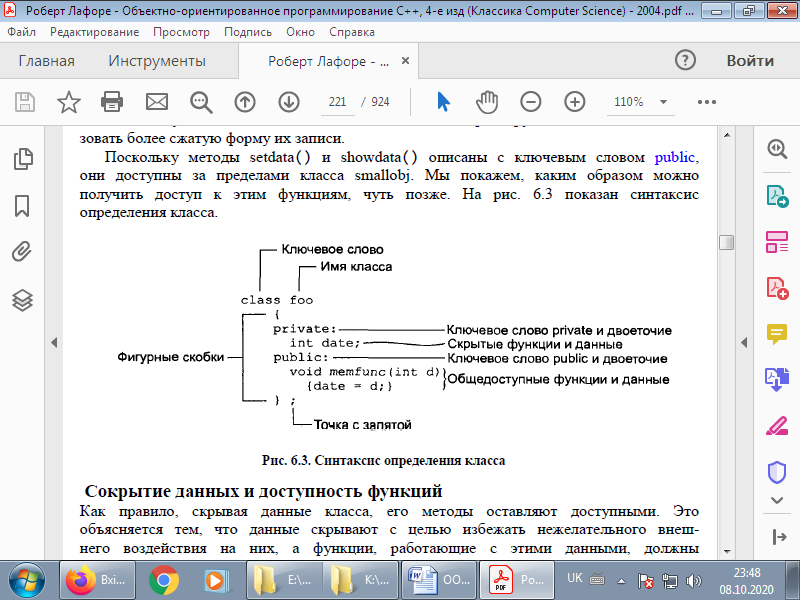


Рис.2 . Синтаксис визначення класу

Б’єрн Страуструп:

"Как правило, класс имеет интерфейс и реализацию. Интерфейс — это часть объявления класса, к которой пользователь имеет прямой доступ. Реализация — это часть объявления класса, доступ к которой пользователь может получить только с помощью интерфейса. Открытый интерфейс идентифицируется меткой **public:**, а реализация — меткой **private:**. Итак, объявление класса можно представить следующим образом:

**class X {** // класс имеет имя X

**public:**

// открытые члены:

// – пользовательский интерфейс (доступный всем)

// функции

// типы

// данные (лучше всего поместить в раздел private)

**private:**

// закрытые члены:

// – детали реализации (используется только членами

// данного класса)

// функции

// типы

// данные

**};**

Члены класса по умолчанию являются закрытыми. Иначе говоря, фрагмент

**class X {**

**int mf(int);**

// . . .

**};**

означает

**class X {**

**private:**

**int mf(int);**

// . . .

**};**

Поэтому

**X x;** // переменная x типа X

**int y = x.mf();** // ошибка: переменная mf является закрытой

// (т.е. недоступной)

Пользователь не может непосредственно ссылаться на закрытый член класса.

Вместо этого он должен обратиться к открытой функции-члену, имеющей доступ к закрытым данным. Например:

**class X {**

**int m;**

**int mf(int);**

**public:**

**int f(int i) { m=i; return mf(i); }**

**};**

**X x;**

**int y = x.f(2);**

Различие между закрытыми и открытыми данными отражает важное различие между интерфейсом (точка зрения пользователя класса) и деталями реализации (точка зрения разработчика класса)."

**Правила створення розділів класу**:

* розділи можуть з’являтися в будь-якому порядку і декілька разів;
* якщо не оголошено жодного розділу, компілятор за замовчуванням оголошує усі елементи закритими;
* розміщати дані-елементи класу у відкритому розділі можна тільки за необхідності. Дані-елементи класу звичайно розміщують у закритому, або захищеному розділі, щоб до них мали доступ функції-члени класу, а також функції класів-нащадків;
* для зміни значень даних (полів) слід використовувати функції-члени класу;
* клас може мати декілька конструкторів, але тільки один деструктор.

  Після того, як клас визначений, і його функції-члени оголошені, ці функції-члени повинні бути визначені (описані). Кожна функція-член може бути описана прямо в тілі класу або після тіла класу. Коли функція-член описується після відповідного визначення класу, імені функції передує ім'я класу і оператор дозволу області дії (::). Оскільки різні класи можуть мати члени з однаковими іменами, оператор дозволу області дії «прив'язує» ім'я члена до імені класу, щоб однозначно ідентифікувати функції-члени даного класу.

**Хорошим стилем програмування** вважається винесення оголошення класу, або інтерфейсу, у файл заголовка, який зазвичай має розширення «.h». Виконання класу, тобто визначення функцій-членів, записується у файл з розширенням «.cpp», який називається реалізацією.

**Хороший стиль програмування** рекомендує використовувати при визначенні класу кожен специфікатор доступу до елементів тільки один раз, що робить програму яснішою і простішою для читання.

Розглянемо приклад використання класу.

*Приклад 11.1.* Навести просту програму з використанням класу.

// P11\_1.CPP — использование класса, структура класса

#include <iostream.h>

#include <conio.h>

class myclass {

int p;                // закрытая переменная no умолчанию

public:

void set\_p(int x);        // прототипы функций-членов класса

int get\_p();

}; //---------- описание функций-членов класса

void myclass::set\_p(int x)

{ p = x; }

int myclass::get\_p( )

{ return p;}

main()

{ myclass ob1, ob2;        // объявление объектов

// вызов функций-членов класса

ob1.set\_p(10);

ob2.set\_p(30);

cout<<"p1= "<<ob1.get\_p()<<" ";

cout<<"p2= "<<ob2.get\_p()<<endl;

getch ();

return 0; }

Результати обчислень: **p1= 10 p2= 30**

У програмі оголошено клас з ім’ям myclass, що має одну закриту змінну р і дві відкриті функції-члени класу set\_p() і get\_p(). Перша з функцій призначена для ініціювання закритої змінної р, а друга — для її повернення. В описі класу оголошено тільки прототипи функцій, їх реалізацію наведено за межами опису. Але якщо функція складається з декількох операторів, її можна розмістити всередині класу. Оскільки ці функції є членами класу, то вони мають доступ до закритої змінної р. У випадку, коли функції-члени описано за межами класу, їхні заголовки повинні складатися з імені класу, операції розширення області класу «::», імені функції та її формальних ар­гументів, якщо вони є, а далі йтиме зміст функції. У головній функції main() спочатку оголошуються два об’єкти типу myclass, потім ці об’єкти ініціюються конкретними значеннями змінної р (ob1.set\_p(10) та ob2.set\_p(30),після цього значення цієї змінної виводяться на екран (cont<<ob1.get\_p(); та cout<<ob2.get\_p0).

***Елементи класу записуються через крапку після імені об’єкта.***

За замовчанням доступ до членів класу – private. Доступ до відкритих членів класу мають будь-які функції в програмі, доступ до закритих членів класу мають тільки члени-функції даного класу і друзі класу. Дані-члени класу зазвичай робляться закритими, а функції-члени – відкритими. Глибокий сенс такого підходу полягає в тому, що дійсне представлення даних усередині класу не торкається клієнтів (користувачів) класу. Клієнти класу використовують клас, не знаючи внутрішніх деталей його реалізації. *Якщо реалізація класу змінюється, інтерфейс класу залишається незмінним і дійсний код клієнта класу не вимагає змін*. Це значно спрощує модифікацію систем.

З того, що дані класу закриті, не витікає, що клієнти не можуть змінювати ці дані. Дані можуть бути змінені функціями-членами або друзями цього класу. Наприклад, щоб дозволити клієнтам класу прочитати закрите значення даних, клас може мати функцію «одержати» (get). Щоб дати клієнтам можливість змінювати закриті дані, клас може мати функцію «встановити» (set). Таким чином, доступ до закритих членів класу здійснюється через відкриті члени-функції даного класу.

Основне завдання відкритих членів полягає в тому, щоб дати клієнтам класу уявлення про можливості (послуги), які забезпечує клас. Цей набір послуг складає відкритий інтерфейс класу. Клієнтів класу не повинно торкатися, яким чином клас виконує їх завдання. Закриті члени класу і опис відкритих функцій-членів недоступні для клієнтів класу. Ці компоненти складають реалізацію класу.

Отже, клас - це структурований тип даних, визначений користувачем. Коли клас визначений, ім'я класу може бути використано для оголошення змінної цього типу , які називають екземплярами класу або об'єктами. Функції, що оголошені усередині класу, викликаються ззовні об'єкта даного класу за допомогою оператора вибору елемента «.» або оператора вибору елемента за допомогою покажчика «->».

Наприклад, об'єкт товар характеризується двома атрибутами: найменуванням і ціною. Необхідно ввести з клавіатури дані про товар і вивести їх на екран. Отже, для об'єкта товар можна оголосити клас, що буде містити два поля даних (найменування товару й ціну), і два методи (уведення даних з клавіатури і вивід даних на екран).

**Зображення класу на мові UML.**

На UML – уніфікованій мові моделювання – клас відображається у вигляді прямокутника, розділеного на три частини. У першій міститься ім'я класу, в другій – атрибути, в третій – методи (рисунок 1).

Клас А

Атрибут 1

Атрибут 2

Атрибут 3

…

Метод 1

Метод 2

Вивід

Рисунок 1 – Відображення класу на UML

**Операція розв’язання видимості ::**

Операція розв’язання видимості :: (scope resolution operator) дозволяє здійснити доступ до глобальної змінної чи функції з блоку, в якому оголошена локальна змінна з тим самим ім’я.

Приклад.

1. Операція розв’язання видимості :: найчастіше застосовується для явного звертання до члена класу поза класом.

**Простір імен - namespace**

1. Простір імен – це спосіб логічного групування
2. Операція розв‘язання видимості :: використовується для звертання до функцій з namespace за іменем поза простором

**Поняття класу і поняття об‘єкту**

**Клас -** це визначений програмістом тип. Оголошення класу задає представлення об'єктів цього класу і набір операцій, які можна застосовувати до таких об'єктів.

Якщо клас – це шаблон, то об’єкт це конкретний представник класу, створений за цим шаблоном.

**X x; // Створюємо змінну х – об’єкт класу (типу) Х**

Об'єкти класу можна створювати та ініціалізувати, знищувати, присвоювати, передавати в якості параметрів функції і повертати як результат виконання функції.

***Ініціалізація не дорівнює Створенню.***

**Члени класу**

У списку членів класу можна оголошувати змінні, функції, класи, перерахування, а також дружні функції та класи.

* ***Не можна двічі оголошувати члена класу в списку членів***
* ***Не можна оголосити в класі змінну та функцію з одним ім'ям.***
* ***Членом класу не може бути об’єкт цього ж класу, але може бути покажчик або посилання на об'єкт того ж класу.***

**Різниця між struct і class**

Члени класу без спеціфікатора доступу за замовчанням є приватними. Члени структур та об'єднань за замовчуванням є публічними

Скотт Майєрс. Правило 22: Объявляйте данные-члены закрытыми

*Контрольні запитання*.

1. Що може містити оголошення класу?
2. Як оголосити клас на мові С++?
3. Де може бути визначена (описана) функція класу?
4. Порівняйте звернення члена класу у функціях класу та поза межами цього класу.
5. Назвіть специфікатори доступу до членів класу.
6. Чи може з'являтися у визначенні класу один і той же специфікатор доступу багато разів?
7. У якому порядку можуть з'являтися специфікатори доступу до членів класу?

**Для самостійного вивчення** *(2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

**Рекомендована література**

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2003. – 461 с. URL: <http://www.ph4s.ru/bookprogramir_1.html>
2. Страуструп Б. Язык программирования С++. — СПб.; М.: Невский диалект — ЗАО “Изд-во “Бином”, 1999.
3. Джейс Либерти Освой самостоятельно С++ за 21 день: 3-е изд. пер. с англ.: Уч. пос. – М.: Издательский дом „Вильямс”, 2001. – 816 с.: ил..

4. Липпман С. Б., Лажойе Ж. Язык программирования С++: Вводный курс. — М.: ДМК, 2001. URL: <http://www.insycom.ru/html/metodmat/inf/Lipman.pdf>

5. Дейтел Х., Дейтел П. Основы программирования на С++. – М.: Бином, 1999. – 1024 с. URL: <http://ijevanlib.ysu.am/wp-content/uploads/2018/03/deytel.pdf>

6. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ . [2-е изд.] / Буч Г. - СПб.: Невский Диалект, 1998. - 560 с.

7. Бадд Т. Объектно-ориентированное программирование в действии. [2-е изд.] – СПб.: Изд-во "Питер". 1997. URL: http://khizha.dp.ua/library/Timothy\_Budd\_-\_Introduction\_to\_OOP\_(ru).pdf