

# Механізм контролю назв

# `static` -конструкції

`static` – конструкція зберігає своє розташування

- **концепція зв'язування** – конструкція має видимість в межах файла; назви мають внутрішнє зв'язування
- **концепція статичного розміщення** – конструкція утворюється один раз за фіксованою адресою в статичній області:
  - глобальні об'єкти розміщуються в статичній області за замовчуванням (без `static` еквівалентно `extern` ), ініціалізуються перед входом в `main()` ;
  - локальні об'єкти ініціалізуються при першому виклику функції і зберігають значення між викликами; за замовчуванням об'єкти вбудованих типів ініціалізуються нулем, невбудованих – конструктором за замовчуванням;

# Класи

- Контроль назв за рахунок інкапсуляції
- **static** поля та методи
  - локалізація в протоколі класу зовнішніх щодо об'єктів полів(змінних класу) і функцій
  - визначення (ініціалізація) полів поза протоколом
  - **this** в метод не передається, з нього можна викликати також лише статичні методи
  - виклик методів через назву класу без існування об'єктів  
**X::f()** ;
- Статичні глобальні об'єкти утворюються відповідними конструкторами до виклику **main()** , знищуються шляхом виклику деструкторів у зворотньому до утворення порядку при виході з **main()** або при явному виклику **exit()**

# Вкладені і локальні класи

- Вкладені класи
  - оголошуються в протоколі іншого класу
  - для використання, як правило, в межах протоколу зовнішнього класу
  - можуть мати статичні елементи
- Локальні класи
  - оголошуються в у блоці функції
  - для використання в блоці оголошення
  - не можуть мати статичних полів

# Простори назв

- Визначення простору назв

```
namespace ідентифікаторopt {  
    оголошення членів простору назв  
}
```

- Визначення простору назв належить або глобальній області видимості, або іншому простору назв
- Визначення простору назв може бути продовженим в інших заголовкових файлах
- За замовчуванням, кожна одиниця компілювання утворює неіменований (анонімний) простір назв, який можна поповнювати визначенням **namespace** без ідентифікатора
- До простору можна доєднати **friend** елементи

# Використання **namespace**

- Оператор області видимості  
(**NS** -ID простору, **X** - ID елемента простору)

**NS :: X**

- Директива

**using namespace NS;**

- усі ідентифікатори, оголошені в , стають доступними до кінця блоку з директивою без оператора області видимості
- конфліктуючі ID кваліфікують
- директиви не використовують у заголовкових файлах

- Оголошення

**using NS :: X;**

- у відповідному блоці стає доступним ID **X**
- може розміщуватись будь-де, де допустиме оголошення
- не має інформації про тип

- Псевдонім всього простору

**namespace NS1=NS;**