

# Bài 6.6: Các kiểu lồng nhau

---

- ✓ Khái niệm và mục đích sử dụng
- ✓ Cú pháp và đặc điểm
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

# Khái niệm và mục đích sử dụng

- ✓ Nested types là các kiểu(struct, class, interface) được định nghĩa bên trong một kiểu khác.
- ✓ Ví dụ nested class là lớp được định nghĩa bên trong một class khác.
- ✓ Các kiểu lồng nhau cho phép ta gom nhóm các lớp chỉ được sử dụng ở một lớp vào nằm gọn bên trong lớp sử dụng nó.
- ✓ Các kiểu lồng nhau do đó làm tăng tính đóng gói dữ liệu, tạo code dễ đọc, dễ bảo trì, nâng cấp.
- ✓ Các lớp được khai báo bên trong 1 lớp khác thì không nên được truy cập trực tiếp vào từ bên ngoài lớp chứa nó để tránh phá vỡ tính đóng gói.
- ✓ Lớp chứa một lớp khác bên trong nó gọi là outer class(lớp bên ngoài).
- ✓ Lớp được định nghĩa bên trong một lớp khác gọi là inner class(lớp bên trong).
- ✓ Mặc định nếu không có access modifier thì inner class sẽ là private.

# Cú pháp và đặc điểm

✓ Cú pháp:

```
2 references
class Student Outer class
{
    private FullName __fullName = new FullName("", "", "");
    2 references
    public string Name ...
    3 references
    private class FullName... Inner class
}
```

- ✓ Inner class có thể được khai báo với 6 access modifier từ private tới private protected.
- ✓ Ta có thể tạo đối tượng của inner class ở trong outer class.
- ✓ Inner class có thể truy cập thành phần static và non-static của outer class.

# Ví dụ

3 references

```
class Student
```

```
{
```

```
    private FullName _fullName = new FullName("", "", "");
```

```
    private int age = 20;
```

2 references

```
    public string Name ...
```

```
    public static float MAX_GPA = 4.0f;
```

0 references

```
    private class InnerClass
```

```
{
```

0 references

```
    public void DoSomething(Student s)
```

```
{
```

```
        Console.WriteLine("MAX_GPA = " + MAX_GPA); // ok
```

```
        Console.WriteLine("Age = " + s.age); // still ok
```

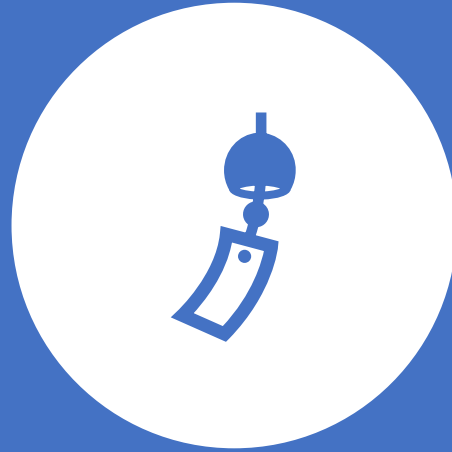
```
}
```

```
}
```

3 references

```
private class FullName...
```

```
}
```



# Nội dung tiếp theo

## Tìm hiểu về record