GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Khoá học: Game Developer

Bài 07: Lớp trừu tượng (Abstract class) và lớp giao diện (Interface)

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

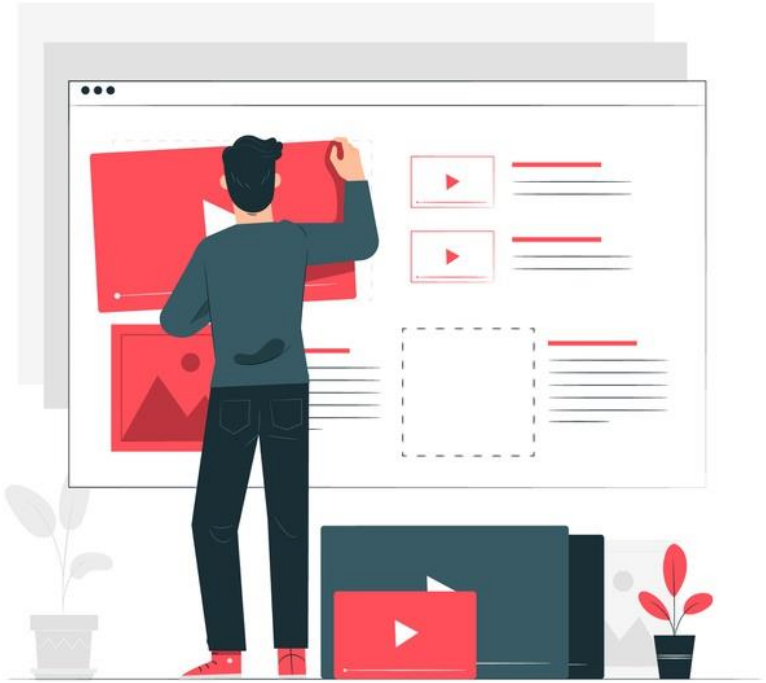
Mục tiêu bài học

Kết thúc bài học này, bạn có thể

- Hiểu khái niệm trừu tượng.
- Xây dựng và hiện thực lớp trừu tượng.
- Xây dựng và hiện thực các phương thức trừu tượng
- Xây dựng và hiện thực lớp giao diện.

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Nội dung



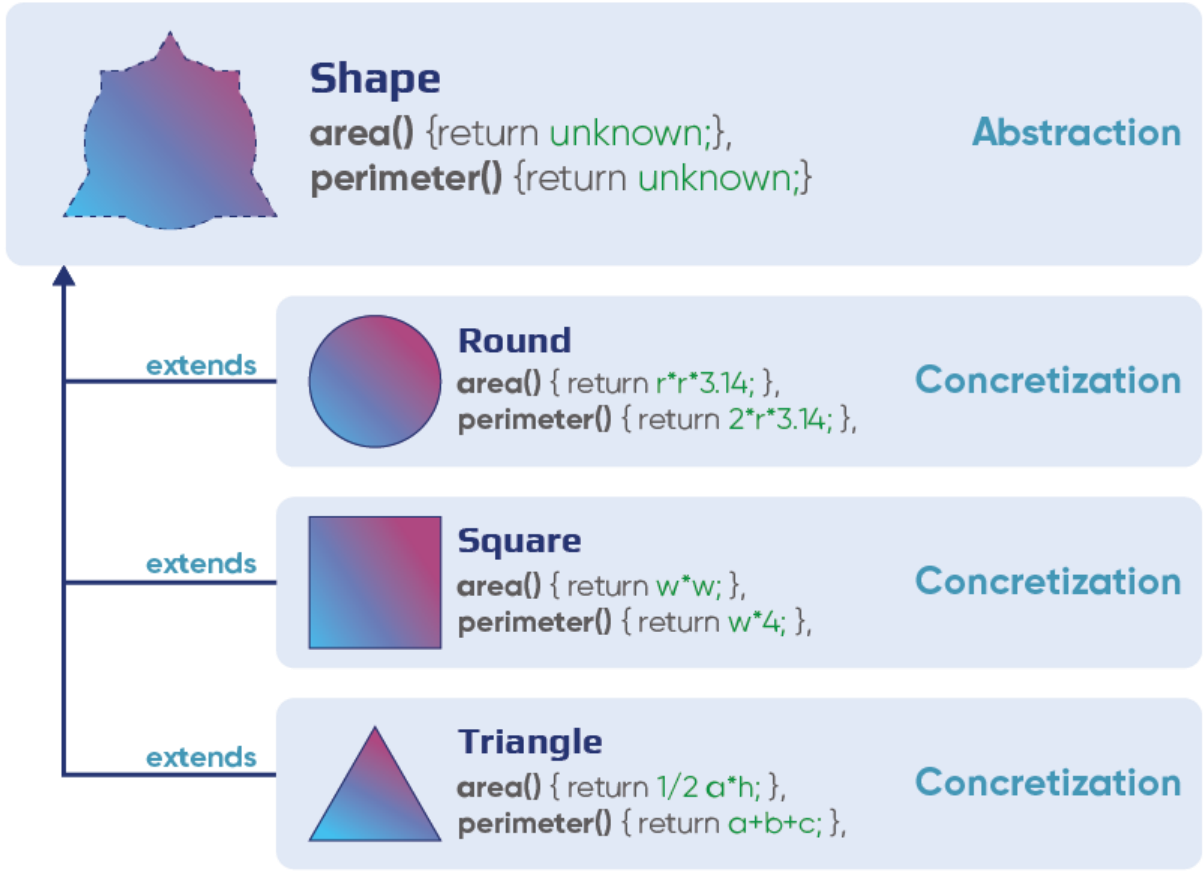
1. Khái niệm Trừu tượng (Abstraction)
2. Lớp giao diện (Interface)
3. Hiện thực lớp giao diện
4. Abstract class và Abstract method

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Sự trừu tượng

Abstraction

- Trong lập trình, khái niệm **trừu tượng** được xem như là một khuôn mẫu không có hình thù xác định, chỉ có khung sườn chứa những hành vi có nội dung chưa được làm rõ.
- Các thành viên hiện thực sự trừu tượng bằng cách **cụ thể hoá** các hành vi.

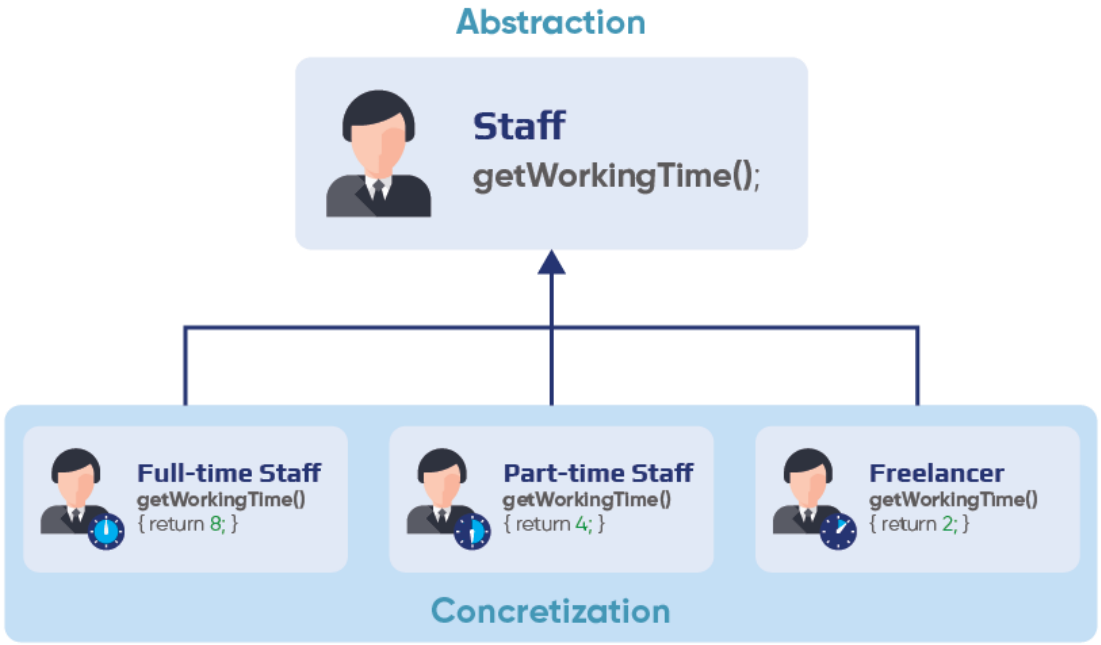


GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Sự trừu tượng

Example

- Tất cả nhân viên trong công ty đều được quy định về thời gian làm việc nhưng chưa biết tính thế nào vì cần biết hình thức làm việc của từng nhân viên thì mới có công thức tính cụ thể.
  - ✓ Nhân viên toàn thời gian làm 8 tiếng/ngày
  - ✓ Nhân viên bán thời gian làm 4 tiếng/ngày
  - ✓ Cộng tác viên làm 2 tiếng/ngày



Vậy **Nhân viên** được xem là **trừu tượng**.

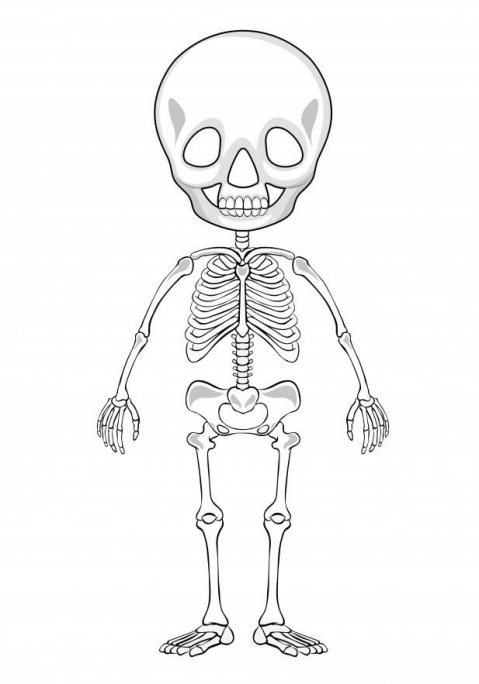
Các nhân viên toàn thời gian, bán thời gian, cộng tác viên là **cụ thể hoá** từ Nhân viên.

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Hình thức thể hiện

sự trừu tượng trong lập trình

- Có 2 hình thức thể hiện sự trừu tượng:
  - Thể hiện bằng **Interface**
  - Thể hiện bằng **Abstract class**
- Sự trừu tượng có thể bao hàm toàn bộ các phương thức cần hiện thực hoặc chỉ một vài phương thức trong một lớp.



Interface

Abstract class

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Interface

Lớp giao diện

- **Interface** có tầm vực truy xuất mặc định là public. Các phương thức trong interface mặc định không có từ khoá public.
- **Interface** chỉ chứa các abstract method và hằng số (final) có tầm vực truy xuất là (public).
- Khi một class cụ thể hoá từ Interface bằng dấu ":". Một class có thể cụ thể hoá từ nhiều Interface – Cách sử dụng đa kế thừa.
- Không thể tạo đối tượng từ Interface vì mọi thứ vẫn còn khái quát (method không có nội dung).

```
interface IStaff {
    int getWorkingTime();
}
```

```
class FullTimeStaff : IStaff {
   public int getWorkingTime()
   {return 8;}
}
```

```
class PartTimeStaff : IStaff {
   public int getWorkingTime()
   {return 4;}
}
```

```
class Freelancer : IStaff {
   public int getWorkingTime()
   {return 2;}
}
```

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Interface

Lớp giao diện

- Các lớp cụ thể hoá Interface thì cần viết lại (override) **toàn bộ** các method trong Interface.
- Các lớp sau khi override tất cả method từ interface vẫn có thể định nghĩa method của riêng lớp đó.

```
interface IStaff {
    int getWorkingTime();
}
```

```
class FullTimeStaff : IStaff {
   public int getWorkingTime() {return 8;}
   public void getAnnualLeave() {…}
}
```

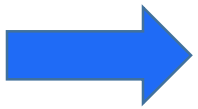
GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Đa kế thừa trong Interface

Multi-inheritance with Interface

```
interface IDaddy {
    void method_1();
    void method_2();
}
```

```
interface IMommy {
    void method_3();
}
```



```
class Child : IMommy, IDaddy {
    public void method_1(){
        System.out.println("I got method 1 from Daddy");
    }
    public void method_2(){
        System.out.println("I got method 2 from Daddy");
    }
    public void method_3(){
        System.out.println("I got method 3 from Mommy");
    }
}
```

GREEN ACADEMY
IT·DESIGN·KOREAN ACADEMY
VIETNAM

Abstract Class

Lớp trừu tượng

- **Về bản chất, lớp trừu tượng cũng là một lớp (class)**, nghĩa là:
  - Abstract Class cũng có fields và method như một class bình thường.
  - Các method bao gồm luôn cả nội dung bên trong thân hàm.
- **Lớp trừu tượng cũng kết hợp với tính trừu tượng của Interface**, nghĩa là:
  - Abstract Class cũng chứa các phương thức không có thân hàm (phương thức trừu tượng) bằng từ khoá abstract.
  - Lớp cụ thể từ lớp trừu tượng vẫn phải hiện thực lại toàn bộ các phương thức trừu tượng.
  - Không thể tạo ra đối tượng từ Abstract Class.
- **Các lớp có chứa phương thức trừu tượng phải được khai báo là lớp trừu tượng.**

```
abstract public class Staff {
    double salaryIndex;
    //abstract method
    abstract public int getWorkingTime();
    //non-abstract method
    public void promote(){
        salaryIndex += 1.5;
    }
}
```

```
class FullTimeStaff implements Staff {
  @Override
  public int getWorkingTime() {return 8;}
  public void getAnnualLeave() {…}
}
```