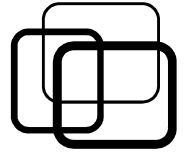


Mẫu Template Method

GV. Nguyễn Minh Huy

cuu duong than cong . com

Mẫu Template Method



■ Ngữ cảnh:

■ Bài toán:

- Xây dựng thuật toán khung.
- Thiết lập event trong lập trình.
- Cài đặt hook sự kiện.

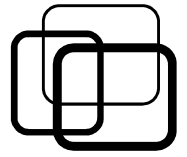
■ Mục tiêu:

- Một quy trình khung cố định.
- Có thể cải tiến, mở rộng các bước trong quy trình.
- ➔ Cải biên một phần phương thức.

cuu duong than cong . com

cuu duong than cong . com

Mẫu Template Method



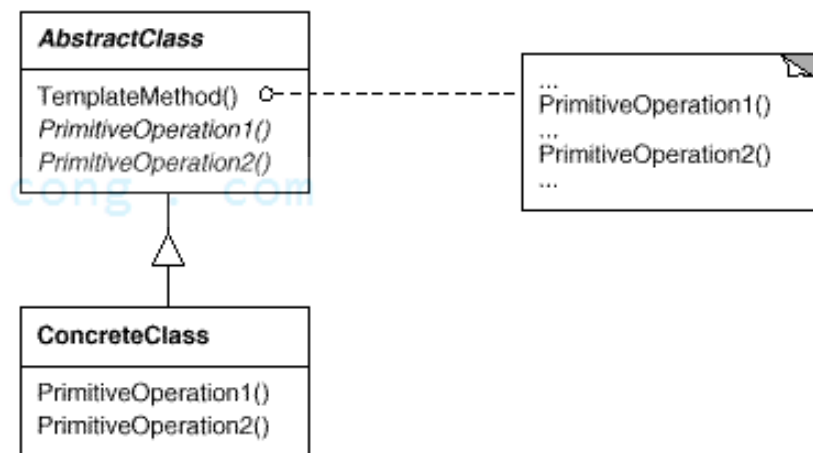
■ Hướng tiếp cận:

■ Cài đặt quy trình khung:

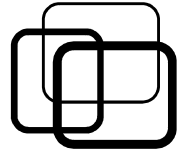
- Dùng phương thức không ảo (non-virtual).

■ Cải tiến, mở rộng các bước trong quy trình:

- Tách các bước cần sửa thành phương thức độc lập.
- Phương thức khung gọi các phương thức này.
- Các lớp kế thừa cải biên lại.



Mẫu Template Method



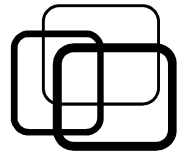
■ Cài đặt:

```
class Form
{
public:
    void load()
    {
        onBeforeLoad();
        // Cài đặt load form...
        onAfterLoad();
    }
    virtual void onBeforeLoad() { }
    virtual void onAfterLoad() { };
};
```

```
class WelcomeForm
{
public:
    void onBeforeLoad()
    {
        // Load ảnh giới thiệu...
    }
}

class LoginForm
{
public:
    void onAfterLoad()
    {
        // Focus ô nhập username...
    }
}
```

Mẫu Template Method



■ Các vấn đề xung quanh:

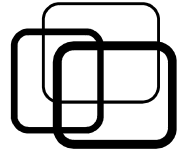
■ Áp đặt quy trình khung với người dùng:

- Chỉ có thể cải biên các bước con.
- Tuân thủ quy trình thực hiện các bước.

➔ Dùng protected...

```
class Form {  
    public:  
        void load() {  
            onBeforeLoad();  
            // Cài đặt load form...  
            onAfterLoad();  
        }  
    protected:  
        virtual void onBeforeLoad() { }  
        virtual void onAfterLoad() { };  
};  
  
void main()  
{  
    Form f = new WelcomeForm();  
  
    f.onAfterLoad();           // Lỗi  
    f.onBeforeLoad();         // Lỗi  
  
    f.load();  
}
```

Mẫu Template Method



■ Các vấn đề xung quanh:

■ Áp dụng giảm trùng lặp mã nguồn:

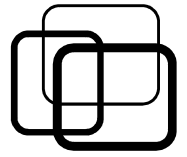
- Phương thức cài biên ở các lớp kế thừa có phần chung.

```
class Animal
{
public:
    virtual void talk() = 0;
};
```

```
class Cat {
public:
    void talk() {
        prepareSound();
        // Cài đặt kêu meo meo...
    }
};
```

```
class Dog {
public:
    void talk() {
        prepareSound();
        // Cài đặt kêu gâu gâu...
    }
};
```

Mẫu Template Method



■ Các vấn đề xung quanh:

■ Áp dụng giảm trùng lặp mã nguồn:

- Tách phần chung thành Template Method.
- Chuyển phần riêng thành Primitive Operation.

```
class Animal
```

```
{
```

```
public:
```

```
    void talk()
```

```
    {
```

```
        prepareSound();
```

```
        doTalk();
```

```
    }
```

```
protected:
```

```
    virtual doTalk() = 0;
```

```
};
```

```
class Cat {
```

```
protected:
```

```
    void doTalk() {
```

```
        // Cài đặt kêu meo meo...
```

```
    }
```

```
};
```

```
class Dog {
```

```
protected:
```

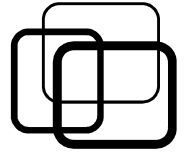
```
    void doTalk() {
```

```
        // Cài đặt kêu gâu gâu...
```

```
    }
```

```
};
```

Mẫu Template Method



■ Các vấn đề xung quanh:

■ Áp dụng tránh quên gọi phương thức cha:

- Phương thức cha cung cấp cài đặt mặc định.
- Phương thức con cải biên và gọi lại cha.

```
class Button {  
public:  
    virtual void onDraw() {  
        // Cài đặt mặc định...  
    }  
};  
  
class CustomButton {  
public:  
    void onDraw() {  
        // Cài đặt vẽ...  
        Button::onDraw();  
    }  
};
```

```
class Button {  
public:  
    virtual void onDraw() {  
        hookOperator();  
        // Cài đặt mặc định...  
    }  
};  
  
class CustomButton {  
protected:  
    void hookOperation() {  
        // Cài đặt vẽ...  
    }  
};
```