

Nguyên tắc Single Responsibility Principle (SRP) và Don't Repeat Yourself (DRY)

- Họ và tên: Phạm Thị Mỹ Hạnh
- MSSV: 22127104

Single Responsibility Principle (SRP) và Don't Repeat Yourself (DRY) là hai nguyên tắc quan trọng trong lập trình phần mềm giúp cải thiện chất lượng mã nguồn, bảo trì dễ dàng và giảm thiểu lỗi.

1. Single Responsibility Principle (SRP)

- Giải thích: Nguyên tắc Single Responsibility Principle (SRP) là mỗi lớp hoặc mô-đun trong phần mềm chỉ nên thực hiện một công việc duy nhất, thay vì làm nhiều công việc khác nhau. Khi một lớp có quá nhiều trách nhiệm, nó sẽ dẫn đến khó bảo trì, mở rộng và dễ dẫn đến lỗi.
- Ví dụ: Ví dụ dưới đây mô tả một class Person nhưng lại bao gồm cả chức năng validate email, và việc validate email này không nên được bao hàm bên trong class Person vì nó không liên quan gì tới hành vi hay method trong class Person.

```
class Person {
  public fullname : string;
  public email : string;
  constructor(fullname : string, email : string){
    this.fullname = fullname;
    if(this.validateEmail(email)) {
      this.email = email;
    }
    else {
      throw new Error("Invalid email!");
    }
  }
  validateEmail(email : string) {
    var re = /^[^\w-]+(?:\.[^\w-]+)*@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$/i;
    return re.test(email);
  }
  greet() {
    alert("Hi!");
  }
}
```

Chúng ta nên tách class phía trên ra thành 2 class là Person và Email để thực hiện hay chức năng khác nhau. Class Email chịu trách nhiệm validate Email.

```
class Email {
    public email : string;
    constructor(email : string){
        if(this.validateEmail(email)) {
            this.email = email;
        }
        else {
            throw new Error("Invalid email!");
        }
    }
    validateEmail(email : string) {
        var re = /^[^\w-]+(?:\.[^\w-]+)*@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$/i;
        return re.test(email);
    }
}

class Person {
    public fullname : string;
    public email : Email;
    constructor(fullname : string, email : Email){
        this.email = email;
        this.fullname = fullname;
    }
    greet() {
        alert("Hi!");
    }
}
```

2. Don't Repeat Yourself (DRY)

- Giải thích: Nguyên tắc Don't Repeat Yourself (DRY) nêu rằng ta không nên lặp lại mã nguồn. Nếu ta viết một đoạn mã giống nhau nhiều lần, điều đó không chỉ làm tăng độ phức tạp mà còn dễ gây lỗi khi bạn cần thay đổi một phần trong mã mà quên cập nhật các phần khác. Thay vì sao chép và dán thì nên tạo các hàm hoặc lớp để tái sử dụng mã một cách dễ dàng hơn.
- Ví dụ:
Ở đoạn code bên dưới việc tính tổng giá của các sản phẩm bị lặp lại và vi phạm nguyên tắc DRY làm lặp lại đoạn code không cần thiết.

```

1  class Order:
2      def calculate_total(self, items):
3          total = sum(item.price for item in items)
4          total_with_tax = total * 1.2
5          return total_with_tax
6
7  class ShoppingCart:
8      def calculate_cart_total(self, items):
9          total = sum(item.price for item in items)
10         total_with_tax = total * 1.2
11         return total_with_tax
12

```

Ta có thể tạo một hàm riêng để tính tổng tiền và thuế trên sản phẩm đầu vào và phần trăm thuế mong muốn và tái sử dụng nó.

```

1  def calculate_sum_plus_tax(items, tax_rate=0.2):
2      total = sum(item.price for item in items)
3      return total * (1 + tax_rate)
4
5  class Order:
6      def calculate_total(self, items):
7          tax_rate = 0.2
8          return calculate_sum_plus_tax(items, tax_rate)
9
10 class ShoppingCart:
11     def calculate_cart_total(self, items):
12         return calculate_sum_plus_tax(items, tax_rate= 0.5)
13

```