**SOLID** là một nguyên tắc phổ biến để thiết kế một dự án tốt nhằm mục đích:

+ dễ mở rộng

+ dễ bảo trì

S.O.L.I.D:

S – Single responsibility principle – nguyên lý trách nhiệm duy nhất

O – open-closd principle – nguyên lý đóng mở

L – Liskov substitution principle – nguyên thế thay thế Liskov

I – interface segregation principle – nguyên lý phân tách interface

D – Dependency Inversion principle – nguyên lý Đảo ngược phụ thuộc

**Single responsibility principle**

Mỗi lớp ( class ) chỉ đảm nhiệm 1 trách nhiệm duy nhất . Điều này giúp việc thay đổi, mở rộng, update, sửa chữa dự án dễ dàng hơn, tránh trường hợp đọc 1 cái class dài cả nghìn dòng, xóa sửa mã nguồn

**Open -closed principle :**

Should be open for extension, but closed for modification

Ví dụ là hàm sort:

Nếu muốn chỉnh sửa, bố sung, mở rộng nhiều loại hàm sort khác nhau. Nên viết ra 1 cái interface SortComparator, sau đó sẽ implement cho từng kiểu sort mà mình muốn

**Liskov subsitution principle**

Trong một chương trình, các object của class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình

**Interface segregation principle:**

Tách các interface để sử dụng khi cần thiết. Không bắt buộc implement một interface nếu không sử dụng đến nó. Không nên implement các phương thức thừa

**Dependency Inversion principle:**

Code như thế này đấy để 2 cái module high và low không phụ thuộc vào nhau, nên phụ thuộc vào abstraction ở giữa. Phải làm cho nó phân tách ra, chỉ phụ thuôc jvào abstraction ở giữa thôi.

Đối tượng Bagel, Znayka, Cư dân Zmeevka, Screwdriver, Gvozdyk, Carnation, dragon

Enum Locations{

GREENCITY,

BEACH,

STREET,

RIVER,

}

Abstract class Human{

Private String name;

Private String Mood;

Public void getName()

Public void getMood()

}

Class Person extends Human{

@override

void describe(){

}

}

Interface Action(){

Void Move()

}

* Biết gì đó ( Bagel)
* Tình nguyện đưa ai đó lên xe (Bagel)
* đi đến thanh phố Xanh (Bagel và Znayka)
* Đi trên đường với khuôn mặt vui vẻ ( cư dân Zmeevka)
* không còn ở đâu đó, cụ thể là làng zmeevka ( con rồng)
* Đã được tìm thấy ( Bagel và Screwdriver)
* Được đi học lại (Gvozdyk)
* Không tin tưởng, Theo dõi Gvozdyk , sợ Gvozdyk làm vỡ kính ( cư dân Zmeevka)
* Nhìn thấy Carnation trên sông ( Người dân Zmeevka)
* Ngồi trên bãi biển, với chiếc quần lót, giặt quần áo ( Carnation)

Có những hành động tiêu biểu nào:

Di chuyển : ( Bagel và Znayka đi đến thành phố xanh , cư dân Zmeevka đi trên đường )