Бублик, узнав о том, что Знайка отправляется в Зеленый город, вызвался отвезти его на своей машине, и скоро они уехали. Жители Змеевки с веселыми лицами ходили по улицам. Все были рады тому, что дракона нет, что Бублик и Шурупчик нашлись, и в особенности тому, что Гвоздик перевоспитался. Некоторые, правда, не верили в это перевоспитание и подозрительно следили за ним, боясь, как бы он опять не начал бить стекла. Через некоторое время Гвоздика увидели на реке. Он сидел на берегу в одних трусиках и стирал свою одежду.

Dịch theo GG dịch:

Baglik, biết rằng Znayka sẽ đến Thành phố Xanh, tình nguyện đưa anh ta lên xe của mình, và họ nhanh chóng rời đi. Những cư dân của Zmeevka đi trên đường với khuôn mặt vui vẻ. Mọi người đều vui mừng vì không có con rồng nào, rằng Bagel và Screwdriver được tìm thấy, và đặc biệt là Gvozdyk đã được học lại. Tuy nhiên, một số người không tin vào cuộc cải tạo này và theo dõi anh ta một cách nghi ngờ, sợ rằng anh ta sẽ lại bắt đầu làm vỡ kính. Một lúc sau, người ta nhìn thấy Carnation trên sông. Anh ấy ngồi trên bãi biển chỉ trong chiếc quần lót và giặt quần áo của mình.

Nhiệm vụ: theo cái đoạn text này, code ra chương trình thể hiện được hết các sự kiện xảy ra.

**Thiết kế 1:** ( hơi tù)

Ta có các **đối tượng**: Đối tượng Bagel, Znayka, Cư dân Zmeevka, Screwdriver, Gvozdyk, Carnation,

Tư tưởng là Mỗi đối tượng sẽ là 1 class kế thừa từ abstract class person đồng thời implements các hành động mà được nhiều đối tượng sử dụng và có riêng cho nó các hành động mà chỉ mình nó thực hiện

Abstract class Person{

+ thuôc tính: Name, mood, Location

+method : getter, setter

+ abstract method SpecialAction() // mỗi đối tượng sẽ có hành động đặc biệt của riêng nó, ta sẽ override cái hành động này

}

Ok, cùng liệt kê các hành động.

* 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)
* 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)
* Liệt kê các hành động đặc biệt mà chỉ có duy nhất một đối tượng thực hiện nó: ( sẽ override từ abstract class)

Ví dụ như: +Bagel : Biết gì đó

+Gvozdyk : Được đi học lại

+Cư dân Zmeevka: Sợ điều gì đó

+ Carnation : giặt quần áo

* Các hành động mà có thể được nhiều đối tượng sử dụng ta sẽ viết thành từng interface rồi implement nó. ( Cụ thể là hành động di chuyển)

Ví dụ như: + Hành động di chuyển ( Bagel và Znayka đi đến thành phố Xanh, Cư dân Zmeevka đi trên đường, Cư dân Zmeevka theo dõi Gvozdyk)

Interface SpeicalMove(){

Void Move() // Các đối tượng nào có hành đông di chuyển thì sẽ implement và override cái method này

}

* Các hành động khác thì ta sẽ tự thêm vào class riêng của đối tượng thực hiện nó.

Ví dụ như Bagel có nhiều hành động:

+ Biết gì đó đã đc override từ absstract,

+ Di chuyển đến thành phố GreenCity ( đã đc implement từ interface Move)

+ Tình nguyện đưa ai đó lên xe ( Hành động này cần thêm vào riêng cho Bagel trong class Bagel)

Ví dụ như Cư dân zmeevka:

+Sợ điều gì đó đã đc override từ abstract class

+Đi trên đường với khuôn mặt vui vẻ ( đc implêmnt từ interface Move)

+ Những hành đồng cần thêm vào riêng cho Zmeevka

* Không tin tưởng vào điều gì đó
* Theo dõi ai đó
* Nhìn thấy ai đó ở đâu

**Vấn đề :** Vì mỗi đối tượng là 1 class nên phải tạo kha khá class

+ Có vẻ như chưa xử lý tốt ở việc tạo ra các hành động tương ứng với từng đối tượng

+ Chưa thấy rõ liệu có vi phạm SOLID hay không?