Design Pattern – Mẫu thiết kế

Trong kỹ thuật phần mềm, mẫu thiết kế là một mô tả được áp dụng linh hoạt nhằm giải quyết một vấn đề trong nhiều tình huống thiết kế phần mềm cụ thể.

Thực tế có rất nhiều mẫu thiết kế từ đơn giản đến phức tạp nhằm hỗ trợ giải quyết các trường hợp khác nhau khi xây dựng mã nguồn chương trình. Tuy nhiên, trong phạm vi bài viết, tôi chỉ đề cập tới một trong những mẫu thiết kế đơn giản nhất đó là Singleton pattern.

Singleton Pattern

Khái niệm và đặc điểm

Singleton là một trong những mẫu thiết kế đơn giản nhất trong các mẫu thiết kế. Đây là mẫu thiết kế được áp dụng nhằm mục đích:

* Xây dựng các lớp Singleton mà mỗi lớp chỉ có duy nhất một thể hiện.
* Có thể truy xuất được các thể hiện duy nhất đó mọi lúc mọi nơi trong chương trình.

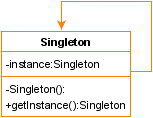
Hiện thực

Khi hiện thực singleton pattern phải đảm bảo rằng chỉ có một thể hiện (instance) duy nhất được tạo ra và thể hiện tại có thể dùng mọi lúc mọi nơi. Nói cách khác, khi xây dựng lớp cần một kỹ thuật để có thể truy xuất được vào các thành viên public của lớp mà không tạo ra một thể hiện nào (thông qua việc khởi tạo bên ngoài lớp).

Để chắc chắn rằng, không có bất kỳ thể hiện nào bên ngoài lớp được tạo ra, constructor của lớp đó sẽ có phạm vi truy cập là private. Điều đó cũng có nghĩa là thể hiện duy nhất được tạo ra thông qua chính lớp mà chúng ta xây dựng. Vì vậy, ta sử dụng thành phần static để thực hiện điều này. Ta tạo một thể hiện của lớp kiểu private static và một refractor trả về đối tượng thuộc chính lớp đó.

Trong C++, singleton định nghĩa một thuộc tính static là một con trỏ trỏ tới thể hiện duy nhất của lớp đó.

Mô hình



Mã nguồn

Khai báo lớp Singleton

1. #include <iostream>
2. using namespace std;
4. class Singleton
5. {
6. private:
7. static Singleton\* m\_instance;
8. Singleton();
9. public:
10. static Singleton\* getInstance();
11. void method();
12. };

Định nghĩa lớp Singleton

1. #include "Singleton.h"
3. // Init static instance
4. Singleton\* Singleton::m\_instance = NULL;
6. Singleton\* Singleton::getInstance()
7. {
8. if (m\_instance == NULL)
9. {
10. m\_instance = new Singleton();
11. }
13. return m\_instance;
14. }
16. void Singleton::method()
17. {
18. cout << "This is singleton parttern";
19. }

Như vậy, thông qua các truy cập Singleton::getInstance() ta luôn có một đối tượng duy nhất, đồng thời có thể sử dụng nó để truy cập mọi thành phần public trong lớp. Ví dụ để gọi phương thức method trong lớp ta dùng Singleton::getInstance()->method().

Trong quá trình làm việc với lập trình hướng đối tượng (OOP), tôi gặp không ít rắc rối trong việc ghi nhớ và quản lý các đối tượng khi chương trình ngày càng nhiều đối tượng. Khi có nhu cầu khởi tạo một đối tượng ngẫu nhiên hoặc khởi tạo nhiều đối tượng trong một vòng lặp thì càng khó khăn hơn. Nhu cầu phát sinh đã khiến tôi tìm ra giải pháp để giải quyết bài toán của mình, mãi sau này tôi mới biết giải pháp đó đã được sinh ra từ lâu và được gọi là Factory Pattern.

## Đối tượng hướng đến

Những lập trình viên đã có kiến thức về OOP và đang tìm hiểu về Design Pattern. Bài viết này tôi dùng ngôn ngữ C++ để làm ví dụ minh họa, các ngôn ngữ hướng đối tượng khác có cách hiện thực hoàn toàn tương tự.

## Đặt vấn đề

Giả sử tôi đang lập trình một game có nhiều loại Monster, tôi có đoạn mã nguồn như sau:  
Đầu tiên chúng ta có lớp Monster trừu tượng và các loại monster kế thừa từ nó:

1. // class abstract Monster
2. class Monster
3. {
4. public:
5. Monster() {};
7. ~Monster() {};
9. virtual void Appear() = 0;
10. };
12. // class monster Lion
13. class Lion : public Monster
14. {
15. public:
16. Lion() {};
18. ~Lion() {};
20. virtual void Appear()
21. {
22. cout << "A Lion has appeared!" << endl;
23. }
24. };
26. //class monster Dragon
27. class Dragon : public Monster
28. {
29. public:
30. Dragon() {};
32. ~Dragon() {};
34. virtual void Appear()
35. {
36. cout << "A Dragon has appeared!" << endl;
37. }
38. };
40. // class monster Giant
41. class Giant : public Monster
42. {
43. public:
44. Giant() {};
46. ~Giant() {};
48. virtual void Appear()
49. {
50. cout << "A Giant has appeared!" << endl;
51. }
52. };

Mỗi khi cần khởi tạo một đối tượng, chúng ta sử dụng toán tử new như thường lệ:

1. Monster \*m\_Lion1 = new Lion();
2. Monster \*m\_Dragon1 = new Dragon();
3. Monster \*m\_Giant1 = new Giant();

Chúng ta cần một đối tượng có thể quản lý việc khởi tạo các đối tượng này, chỉ cần truyền vào yêu cầu, thì đối tượng đó sẽ trả về đối tượng tương ứng. Để làm được điều đó chúng ta cần sử dụng một đối tượng Factory.

## Sử dụng Factory Pattern

Factory, đúng nghĩa là một nhà máy, và nhà máy này sẽ “sản xuất” các đối tượng theo yêu cầu của chúng ta. Hãy khai báo một enum như một tập hợp các yêu cầu đó:

1. enum eMonster
2. {
3. LION,
4. DRAGON,
5. GIANT,
6. MONSTER\_COUNT
7. };

Bắt tay xây dựng nhà máy nào:

1. class MonsterFactory
2. {
3. public:
4. MonsterFactory() {};
6. ~MonsterFactory() {};
8. static Monster \*createMonster(eMonster \_MonsterID)
9. {
10. switch (\_MonsterID)
11. {
12. case LION:
13. {
14. return new Lion();
15. break;
16. }
17. case DRAGON:
18. {
19. return new Dragon();
20. break;
21. }
22. case GIANT:
23. {
24. return new Giant();
25. break;
26. }
27. default:
28. {
29. return NULL;
30. break;
31. }
32. }
33. }
34. };

Sử dụng Factory:

1. int main()
2. {
3. Monster \*m\_Lion1 = MonsterFactory::createMonster(eMonster::LION);
4. Monster \*m\_Dragon1 = MonsterFactory::createMonster(eMonster::DRAGON);
5. Monster \*m\_Giant1 = MonsterFactory::createMonster(eMonster::GIANT);
7. m\_Lion1->Appear();
8. m\_Dragon1->Appear();
9. m\_Giant1->Appear();
11. return 0;
12. }

Kết quả xuất ra màn hình:  
**A Lion has appeared!  
A Dragon has appeared!  
A Giant has appeared!**

Như vậy, mỗi khi có tạo một class Monster mới, ta lại thêm nó vào trong Factory để quản lý việc khởi tạo dễ dàng hơn.

## Ứng dụng của Factory Pattern

Ngoài việc giúp dễ dàng quản lý việc khởi tạo các đối tượng, thì như đã nói ở phần Tiền đề bài viết, Factory Pattern còn giúp tôi khởi tạo một đối tượng ngẫu nhiên và dùng vòng lặp sinh ra một mảng các đối tượng.

### Khởi tạo đối tượng ngẫu nhiên

Chúng ta sẽ sử dụng hàm Random mà tác giả [Amy Lê](http://www.stdio.vn/users/index/70/amy-le) đã giới thiệu ở bài viết [Random Số Trong C++](http://www.stdio.vn/articles/read/67/random-so-trong-c):

1. int main()
2. {
3. srand(time(NULL));
4. eMonster RandomMonsterID = eMonster(rand() % eMonster::MONSTER\_COUNT);
5. Monster \*RandomMonster = MonsterFactory::createMonster(RandomMonsterID);
7. RandomMonster->Appear();
9. return 0;
10. }

Kết quả xuất ra màn hình tại thời điểm tôi chạy đoạn code trên là:  
**A Dragon has appeared!**  
Mỗi lần chạy sẽ ra một kết quả khác nhau, hoàn toàn ngẫu nhiên.

Giới thiệu

Nếu bạn từng nghe về Design Pattern, bạn có bao giờ thắc mắc chúng là cái gì? Và tại sao chúng lại được nhắc nhiều đến vậy? Đừng lo lắng, tôi hi vọng rằng bài viết này sẽ giúp bạn giải quyết những thắc mắc ban đầu của mình. Với những ai đã biết về Design Pattern, có thể dùng bài viết này để tham khảo và trau dồi kiến thức.

Tiền đề bài viết

Tôi có được may mắn tiếp cận vấn đề này vào hai năm trước, trong lúc nghe một ai đó nói về tác dụng của nó, sự tò mò trong tôi thôi thúc mình tìm hiểu vấn đề. Và hôm nay, khi đã đi được một quãng đường, tôi quyết định chia sẻ lại kiến thức này với các bạn, cũng là cách mà tôi lưu lại những kiến thức của mình.

Đối tượng hướng đến

Đối tượng hướng đến là đọc giả có kiến thức cơ bản về một ngôn ngữ lập trình nhất định và đã có kiến thức về phương pháp lập trình hướng đối tượng. Trong bài viết này, tôi sử dụng ví dụ với ngôn ngữ lập trình C++, các ngôn ngữ khác như Java, C#, Ruby… đều có nguyên lý thực hiện tương tự.

Quan điểm cá nhân về khái niệm Design Pattern

Design Pattern là một khái niệm để nói về các phương pháp giải quyết vấn đề trong lập trình, được các lập trình viên ghi lại, hoặc truyền đạt lại dưới một hình thức nào đó, với mục đích giải quyết nhanh gọn, và lưu trữ cho thế hệ về sau có thể xử lý nhanh vấn đề. Trải qua thời gian, các Pattern ngày càng phong phú về thể loại cũng như hình thức tiếp cận để giải quyết vấn đề.

Chúng ta sẽ tiếp cận Design Pattern đầu tiên có tên là Abstract Factory. Lý do tôi chọn Design Pattern này làm bài viết đầu tiên bởi sự **đơn giản**, và **dễ nắm bắt** nếu bạn đã từng học qua phương pháp lập trình hướng đối tượng.

Abstract Factory Design Pattern

Abstract Factory là sự mở rộng của tính chất đa hình trong lập trình hướng đối tượng. Mục tiêu hướng đến là có thể tạo ra các đối tượng mà chưa biết trước chính xác kiểu dữ liệu của chúng.

Hãy tưởng tượng, Abstract factory là một nhà máy lớn chứa nhiều nhà máy nhỏ, trong các nhà máy đó có những xưởng sản xuất, các xưởng đó tạo ra những sản phẩm khác nhau. Điều này có vẻ trừu tượng, nhưng đừng vội lo lắng, hãy đi đến cuối nội dung của bài này, và ngẫm lại điều tôi vừa nói, nó sẽ giúp bạn nhớ lâu hơn tư tưởng thiết kế này.

Mục tiêu mà Abstract Factory hướng tới

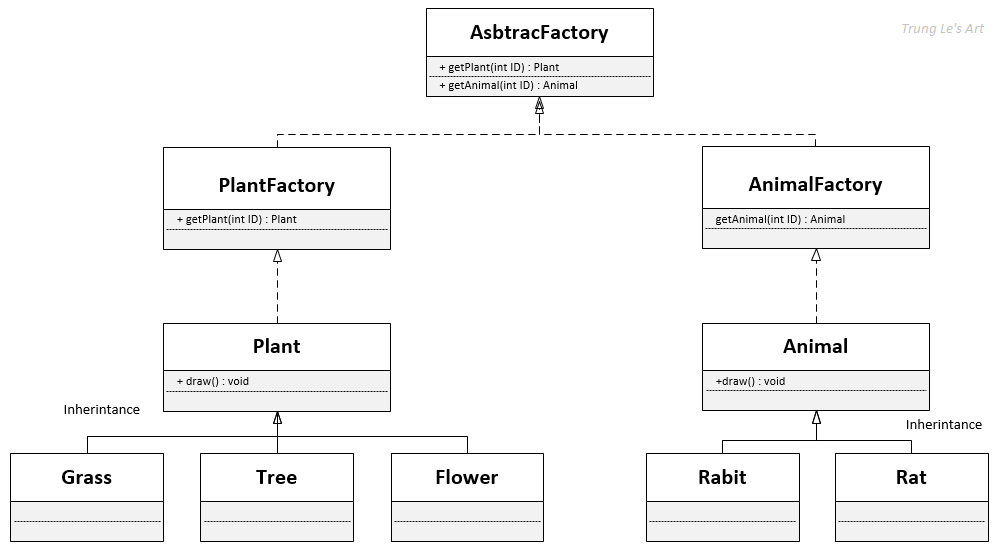
* Tạo ra đối tượng mà không cần biết chính xác kiểu dữ liệu.

*Tôi giải thích điều này cho các bạn rõ ràng hơn. Ví dụ, bạn sẽ đọc dữ liệu từ một file txt để lấy thông tin tạo đối tượng. Trong file txt đó chứa một số nguyên. Trong mã nguồn bạn viết rằng, nếu dữ liệu đọc được là số 1 thì tạo ra đối tượng A, nếu là 2 thì tạo ra đối tượng B. Vậy rõ ràng, đối tượng của bạn chỉ được tạo ra trong quá trình run time, nghĩa là trong lúc bạn viết mã nguồn chương trình, bạn chưa xác định được đối tượng mình cần tạo. Abstract Factory có thể giúp bạn giải quyết vấn đề này.*

* Giúp mã nguồn của bạn trở nên dễ dàng bảo trì nếu có sự thay đổi.

*Trước khi tìm hiểu về điều này, bạn có thể tìm hiểu thêm về*[*Factory Design Pattern*](http://www.stdio.vn/articles/read/275/design-pattern-factory-pattern)*, Abstract Factory được hình dung như là một nhà máy chứa nhiều nhà máy con. Mỗi nhà máy con là một factory. Bạn có thể sử dụng Abstract Factory để tạo ra tất cả các đối tượng trong chương trình của bạn. Khi bạn sửa đổi hoặc thêm mới một đối tượng, các đối tượng khác sẽ không bị ảnh hưởng.*

Mô tả tư tưởng



Sử dụng Asbstract Factory để tạo ra tất cả các đối tượng trong chương trình của bạn.

Hiện thực hóa

Đầu tiên bạn tạo một hệ thống các lớp đối tượng như trên. Bạn cần thực hiện đến cấp Plant. Nghĩa là bạn sẽ tạo ra lớp Grass, Tree, Flower, những lớp này kế thừa từ lớp thuần ảo Plant. Tương tự điều đó với Rabit và Rat. Tôi hi vọng điều này không quá khó với bạn. Nếu gặp rắc rối, hãy xem lại những bài viết về phương pháp lập trình hướng đối tượng trong [***STDIO***](http://www.stdio.vn/)***.***

Sau đó, hiện thực Plant Factory như sau.

1. Plant\* getPlant(int ID)
2. {
3. switch (ID)
4. {
5. case PLT\_GRASS:
6. return new Grass();
7. case PLT\_TREE:
8. return new Tree();
9. case PLT\_FLOWER:
10. return new Flower();
11. default:
12. break;
13. }
14. }

Hiện thực Animal Factory

1. Animal\* getAnimal(int ID)
2. {
3. switch (ID)
4. {
5. case ANL\_RABIT:
6. return new Rabit();
7. case ANL\_RAT:
8. return new Rat();
9. default:
10. break;
11. }
12. }

Hiện thực AbstracFactory

1. class AbstracFactory
2. {
3. public:
5. Plant\* createPlant(int ID)
6. {
7. return getPlant(ID);
8. }
10. Animal\* createAnimal(int ID)
11. {
12. return getAnimal(ID);
13. }
14. };

Một ví dụ để gọi factory trong hàm man().

1. int main()
2. {
3. AbstracFactory\* factory = new AbstracFactory();
5. Plant\* obj1 = factory->createPlant(PLANT::PLT\_GRASS);
6. Animal\* obj2 = factory->createAnimal(ANIMAL::ANL\_RABIT);
8. return 0;
9. }

Một số chú ý

* Trong code mẫu, thay vì hiện thực các factory con là lớp, tôi đã linh động hiện thực thành các hàm, bởi chúng có thể thực hiện cùng một mục đích, nhưng hàm thì không cần thông qua thể hiện, hay tạo lớp static, vấn đề này liên quan đến hiệu suất chương trình nên tôi xin không đề cập sâu.
* Với những ngôn ngữ khác như Java, việc đa kế thừa là không thể, nên bạn có thể thay đổi các lớp factory thành các hàm như trên hoặc thành các interface.
* Bạn có thể kết hợp [Singleton Design Pattern](http://www.stdio.vn/articles/read/263/singleton-pattern) cùng với Abstract Factory để làm cho Abstract Factory có thể gọi được ở mọi nơi trong chương trình. Điều này sẽ linh động trong việc tạo ra các thể hiện. Tuy nhiên bạn cần cân nhắc bởi nó sẽ tiêu tốn nhiều tài nguyên để gọi hàm.

Giới thiệu

Quá trình phát triển của tư duy lập trình cũng tương tự như quá trình phát triển của loài người, đi từ thấp lên cao. Ban đầu là lập trình tuần tự, sau đó là lập trình cấu trúc, rồi lập trình thủ tục, và giờ đây là lập trình hướng đối tượng.

Trải qua một thời kì tư duy lập trình hướng đối tượng trở nên phổ biến, thì những người lập trình tiên phong đã tìm ra các vấn đề chung trong hướng đối tượng và nghĩ ra các giải pháp để giải quyết các vấn đề đó một cách tối ưu trong phần lớn các trường hợp, và những người đó được biết đến dưới cái tên *Gang of Four (GoF)*, những người đã lần đầu tạo nên khái niệm *design pattern* qua cuốn sách *“Design Pattern: A Element of reusable object oriented software”*.

Trong bài viết này sẽ đề cập đến một *design pattern* được sử dụng rất phổ biến trong lập trình hướng đối tượng, đó là *Adapter Pattern*.

Tiền đề bài viết

Ban đầu mục tiêu của bài viết là nhằm để tham gia cuộc thi. Nhưng với sự đam mê chia sẻ kiến thức của mình thông qua những *“dòng chữ trên trang giấy”*. Tác giả đã dành rất nhiều tâm huyết của mình để tạo ra một bài viết mang lại giá trị thật sự, cho cả người đọc lẫn tác giả, giúp người đọc có thể nắm rõ và vận dụng được *Adapter Pattern* trong các project ngoài thực tế.

Đối tượng hướng đến

Bài viết dành cho những lập trình viên đã quen với tư duy lập trình hướng đối tượng, cũng như đã hiểu rõ các đặc trưng và tính chất của hướng đối tượng.

Tổng quan

Lập trình hướng đối tượng là một phương pháp tư duy lập trình mới (so với lập trình hướng thủ tục và lập trình hướng cấu trúc) giúp chúng ta có thể giải quyết tốt hơn các bài toán trong thực tế nhờ khả năng trừu tượng hóa các đối tượng ngoài thực tế vào chương trình thông qua khái niệm *“Lớp”* và *“Đối tượng”*.

Khi lập trình hướng đối tượng ra đời thì khái niệm *“cấu trúc chương trình”* được đẩy lên một tầm cao mới. Một chương trình không còn là một sự thực thi tuần tự từng câu lệnh, hay sự gọi tuần tự các hàm, mà giờ đây, nó là *“các hành động của đối tượng”* và *“sự tương tác giữa các đối tượng”*. Vì lẽ đó mà tư duy lập trình cũng như cấu trúc chương trình của chúng ta cần phải thay đổi.

Trong lập trình hướng đối tượng, một khái niệm cực kì quan trọng đó là *“giao diện của lớp”* (không phải kiểu interface của Java hay C#). *“Giao diện của lớp”* biểu thị những *“tính chất”* bên ngoài của lớp, gồm các hành động của lớp (method) và các thuộc tính của lớp (attribute/property). Cách duy nhất để chúng ta tác động vào một đối tượng là phải thông qua *“giao diện của nó”* (ví dụ: khi muốn đối tượng Cat thực hiện hành động Run thì ta phải thông qua giao diện của lớp Cat, giao diện của nó phải có một phương thức là Run(), khi đó thì ta sẽ gọi Cat.Run()).

Ta có thể coi người sử dụng đối tượng có hai loại: Người định nghĩa (Người định nghĩa ra đối tượng đó) và Người sử dụng (Người sử dụng đối tượng, gọi các phương thức của đối tượng đó). Nếu đứng ở góc nhìn của một *“người sử dụng đối tượng”*, thì chỉ có hai điều ta cần phải quan tâm, một là *giao diện của đối tượng* là gì, và hai là đối tượng đó *thực hiện những hành động* gì. Ta quan tâm tới *“hành động của đối tượng là gì”* vì đó là lý do ta dùng đến đối tượng đó. Còn ta quan tâm tới *"giao diện của đối tượng"* vì ta phải biết giao diện của nó ta mới có thể tương tác được với nó (phải biến đối tượng có những phương thức nào mới có thể gọi, tương tự như khi bạn sử dụng một lớp thì bạn cần phải biết lớp đó có những thuộc tính, phương thức nào). Nếu ta coi một đối tượng trong lập trình là một đối tượng ngoài đời thực thì giao diện của đối tượng sẽ là vẻ bề ngoài của đối tượng (attribute/property) và các hành vi cử chỉ của đối tượng đó (method).

Tầm quan trọng của giao diện

Giao diện của lớp cung cấp cho chúng ta cách thức mà chúng ta dùng để tác động vào đối tượng. Vì lẽ đó mà *“giao diện của lớp”* là một khái niệm rất quan trọng, nó ảnh hưởng rất lớn đến cấu trúc của một chương trình cũng như khả năng bảo trì và nâng cấp của chương trình.

Khi ta nói đến cấu trúc của một cái gì đó, thì nghĩa là ta đang nói đến sự sắp đặt, trật tự cũng như *“giao diện”* (bề ngoài) của các thành phần trong nó. Để những thành phần có thể được ghép nối lại với nhau thì nó phải có *“giao diện”* tương đồng nhau (có thể ghép nối được). Do vậy, khi muốn chỉnh sửa hay nâng cấp một thành phần trong cấu trúc của một cái gì đó, ta phải bảo toàn *“giao diện”* của nó (để nó vẫn có thể ghép vào được các thành phần cũ).

Trong lập trình phần mềm và hệ thống cũng như vậy, cấu trúc của một chương trình là sự sắp đặt, trật tự cũng như giao diện của các đối tượng trong chương trình. Khi ta muốn chỉnh sửa hay nâng cấp một thành phần nào đó của chương trình (đối tượng), để không phải chỉnh sửa thêm các thành phần khác nữa, thì đối tượng được chỉnh sửa/nâng cấp phải bảo toàn được giao diện ban đầu của nó. Đó là lý do giao diện là một khái niệm rất quan trọng đối với cấu trúc của một chương trình. Nếu chương trình chúng ta có cấu trúc tốt, thì nếu bị một lỗi nào đó, dù lớn, chúng ta chỉ cần chỉnh sửa một thành phần/đối tượng nào đó, còn không thì chúng ta phải rà soát lại toàn bộ source code của chương trình để chỉnh sửa lỗi dù lỗi đó có nhỏ đến đâu.

Đặt vấn đề

Có bao giờ bạn gặp phải trường hợp phải port một game hay ứng dụng từ nền tảng này sang nền tảng khác?

Đó là một điều không phải hiếm đối với ngày nay, khi mà sự đa dạng các nền tảng phát triển cũng như nền tảng chạy ứng dụng được tăng nhanh hơn bao giờ hết. Và đây cũng là việc mà bạn thường phải đối mặt khi làm trong một công ty, tham gia các cuộc thi hay ngay cả khi bạn đang ngồi trên ghế nhà trường loay hoay với các đồ án. Vậy việc port một ứng dụng dễ hay khó? Nó sẽ dễ nếu bạn biết cách làm, còn nó sẽ là một công việc cực kì nhức óc nếu bạn chỉ làm theo cách mà bạn nghĩ, không theo bất kì phương pháp nào.

Ban đầu phương pháp mà bạn nghĩ đến ắt hẳn là viết lại ứng dụng đó trên nền tảng khác, và bạn nghĩ đây là phương pháp duy nhất, vì hai nền tảng khác nhau thì không thể dùng lại code viết trên nền tảng này xài cho nền tảng kia được? Nhưng có một điều khá là hiển nhiên nếu bạn để ý, phương pháp đầu tiên mà ta nghĩ ra khi đối mặt với một vấn đề hoàn toàn mới sẽ là một phương pháp tổng quát, và cũng là phương pháp tồi tệ, nhức óc và tốn nhiều công sức nhất.

Vậy đâu là phương pháp tốt nhất trong trường hợp này? Hiển nhiên là phải làm sao đó để ta có thể dùng lại những đoạn mã nguồn mà ta đã viết. Mã nguồn của một chương trình hay game có thể chia làm hai thành phần: thành phần thứ nhất là framework, thành phần thứ hai là mã nguồn mà ta viết. Vì thành phần thứ hai phụ thuộc hoàn toàn vào framework bên dưới, nên khi thay framework thì mã nguồn của thành phần thứ hai hoàn toàn không còn hoạt động được nữa. Vậy cách giải quyết cho vấn đề này rất đơn giản: *tạo cho framework mới một giao diện giống hệt framework cũ!*

*"Giả sử bạn muốn port game từ XNA sang Java. Bên XNA bạn dùng đối tượng SpriteBatch để vẽ, thì qua bên Java bạn cũng tạo một lớp SpriteBatch có giao diện hoàn toàn giống với lớp này bên XNA, như vậy tất cả đoạn code vẽ hình bằng đối tượng SpriteBatch của bạn trong mã nguồn sẽ chạy được!"*

Và đó cũng chính là một mẫu thiết kế trong *design patterns*: **Adapter Pattern** – chuyển đổi giao diện ban đầu của một lớp thành một giao diện phù hợp với sử dụng.

Adapter Pattern

Design Pattern là gì?

*Design pattern* là các mẫu hay mô hình trong thiết kế phần mềm/hệ thống giúp chúng ta giải quyết các vấn đề thường gặp trong lập trình. *Design pattern* cung cấp cho chúng ta các cách thiết kế chương trình để giải quyết tối ưu các bài toán mà ta hay gặp phải trong lập trình như: khởi tạo đối tượng, cấu trúc chương trình, và hành động của các đối tượng…

Adapter trong thực tế

Adapter là một khái niệm rất thông dụng trong đời sống hàng ngày. Ta thường hay bắt gặp các loại adapter như: power adapter (chuyển đổi điện áp), laptop adapter (bộ sạc của laptop) hay memory card adapter… Các adapter này có nhiệm vụ chính là làm cầu nối trung gian để giúp hai đồ vật gì đó có thể hoạt động với nhau.

Ví dụ như laptop không sử dụng nguồn điện xoay chiều 224V, nên để laptop có thể sử dụng được nguồn điện 224V cần có một adapter làm cầu nối trung gian để chuyển nguồn điện xoay chiều 224V thành nguồn điện 1 chiều 12V. Một ví dụ khác là thẻ nhớ, trên thị trường có rất nhiều loại thẻ nhớ nhưng loại thịnh hành nhất ngày nay vẫn là loại micro-SD vì tính nhỏ gọn và phổ biến của nó, vậy nếu bạn có một thẻ micro-SD và một máy ảnh sử dụng thẻ SD, làm sao để có thể cắm thẻ micro-SD này vào máy ảnh? Khi đó ta sẽ sử dụng một adapter để chuyển *“bề ngoài”* của thẻ micro-SD thành SD để có thể cắm vào máy ảnh.

Adapter trong hướng đối tượng

Một trong những mục đích của hướng đối tượng là để giúp chúng ta có thể *“phản ánh”* (hay ánh xạ) tốt hơn các đối tượng ngoài thực tế vào trong lập trình. Vì lẽ đó mà ngoài thực tế có adapter thì trong hướng đối tượng cũng có thể có adapter với mục đích tương tự như adapter ngoài thực tế. Lấy lại ví dụ về một thẻ micro-SD và máy ảnh, nếu ta coi vẻ bề ngoài và kích thước của thẻ nhớ là một phần “giao diện” của thẻ nhớ, thì adapter thẻ nhớ đóng vai trò là một cầu nối trung gian để chuyển đổi *“giao diện”* của thẻ nhớ sao cho nó có thể phù hợp với máy ảnh.

Giả sử ta có một hệ thống phần mềm (hãy tưởng tượng nó là cái máy ảnh), và có một số đối tượng (hãy tưởng tượng nó là cái thẻ micro-SD) được một số developer viết ra nhưng có giao diện không phù hợp với hệ thống này, cách tốt nhất để có thể “ráp” các đối tượng này vào được hệ thống là tạo các adapter để chuyển đổi giao diện của các đối tượng này.

Ý tưởng của Adapter

Như ta đã biết, giao diện của một lớp rất quan trọng, vì nó cho chúng ta biết cách thức mà chúng ta sẽ tương tác với đối tượng. Trong một số trường hợp, giả sử ta đã có sẳn một đối tượng, đối tượng đó là đối tượng ta cần, nhưng lại có giao diện không phù hợp với hệ thống của chúng ta nên ta không thể sử dụng được. Vậy phải chăng ta phải định nghĩa lại đối tượng đó với giao diện khác?

Nếu bạn nghĩ vậy, hãy quay về với thế giới mà hướng đối tượng được sinh ra để tìm câu trả lời. Hãy tưởng tượng chiếc xe máy của bạn là một đối tượng, và vẻ bề ngoài của nó chính là giao diện, giả sử chiếc xe của bạn nhìn khá “bụi bặm”, nhưng khi đi làm thì ông sếp không thích giao diện của chiếc xe đó, và yêu cầu bạn phải đổi một chiếc xe có giao diện thanh nhã và lịch sự hơn. Vậy bạn sẽ làm như thế nào? Đổi xe hay đổi lớp vỏ (giao diện) của nó? Tất nhiên câu trả lời sẽ là đổi lớp vỏ. Hay nói cách khác, câu trả lời cho câu hỏi mà bạn tìm “không ra” trong thế giới hướng đối tượng là: *ta chỉ cần đổi giao diện cũ của đối tượng đó thành giao diện phù hợp, thay vì phải định nghĩa lại nguyên cả đối tượng đó.*

Ý tưởng của **Adapter Pattern** rất đơn giản. Khi bạn có một số các đối tượng, nhưng các đối tượng đó lại không sử dụng được với phần mềm bạn đang xây dựng vì nó có giao diện không phù hợp, vậy giải pháp phù hợp nhất sẽ là tạo nên giao diện mới cho các đối tượng đó bằng một lớp giao diện trung gian mà người ta gọi là *“Adapter”*. Lớp này chỉ có trách nhiệm chuyển đổi giao diện, các phương thức của nó sẽ gọi lại các phương thức của đối tượng mà nó *“adapt”*.

Các khái niệm

Để hiểu về sơ đồ mô tả **Adapter Pattern** thì trước hết bạn phải hiểu về 3 khái niệm:

* Client: Đây là lớp sẽ sử dụng đối tượng của bạn (đối tượng mà bạn muốn chuyển đổi giao diện).
* Adaptee: Đây là những lớp bạn muốn lớp Client sử dụng, nhưng hiện thời giao diện của nó không phù hợp.
* Adapter: Đây là lớp trung gian, thực hiện việc chuyển đổi giao diện cho Adaptee và kết nối Adaptee với Client.

Phân loại Adapter Pattern

Trong hướng đối tượng có hai khái niệm quan trọng song hành cùng nhau, đó là

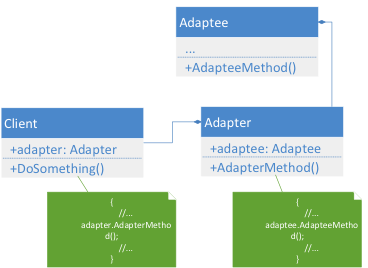
* *Composition*: cấu thành. Nghĩa là một lớp B nào đó sẽ trở thành một thành phần của lớp A (một field trong lớp A). Tuy lớp A không kế thừa lại giao diện của lớp B nhưng nó có được mọi khả năng mà lớp Bcó.
* *Inheritance*: kế thừa. Nghĩa là một lớp Derived sẽ kế thừa từ lớp Base và thừa hưởng tất cả những gì lớp Base có. Nhờ kế thừa mà nó giúp tăng khả năng sử dụng lại code, tăng khả năng bảo trì và nâng cấp chương trình. Và do vậy kế thừa là khái niệm trọng tâm trong hướng đối tượng. Nhưng nó có một nhược điểm, đôi khi nếu chúng ta quá lạm dụng nó, nó sẽ làm cho chương trình của chúng ta phức tạp lên nhiều, điển hình là trong lập trình game. Do vậy đôi lúc trong lập trình game người ta thường có khuynh hướng thích sử dụng *composition*hơn.

Và ứng với hai khái niệm này sẽ cho ta hai cách để chúng ta cài đặt lớp adapter: *Object Adapter và Class Adapter*.

Object Adapter Pattern

Đây là một phương pháp cài đặt **Adapter Pattern** dựa trên ý tưởng về *composition*. Lớp Adapter của chúng ta sẽ chứa một instance của lớp Adaptee bên trong nó. Và giao diện của nó sẽ tương ứng với giao diện mà chúng ta cần. Nhưng các phương thức trong giao diện đó không được định nghĩa, mà nó chỉ gọi lại các phương thức tương ứng của lớp Adaptee. Adapter lúc này chỉ như một cầu nối trung gian giữa Clientvà Adaptee, mục đích để*“chuyển đổi tên phương  thức”* mà Client gọi trên đối tượng Adapter thành phương thức tương ứng ở lớp Adaptee.

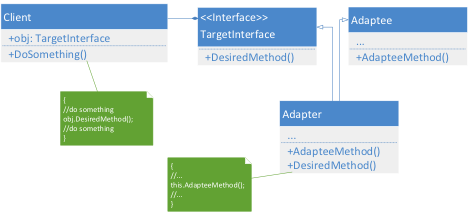
Hay ta có thể hiểu đơn giản, đối với *Object Adapter Pattern*, ta sẽ sử dụng một đối tượng Adapter để “mặc” (wrap) cho đối tượng Adaptee một “bộ áo mới” (giao diện mới). Nếu tưởng tượng Adaptee là một người, thì Adapter chỉnh là bộ đồ, khi Adaptee “mặc” lên mình Adapter thì Adaptee (lúc này là người) sẽ có một vẻ bề ngoài (giao diện) khác.



Class Adapter Pattern

Đây là một phương pháp cài đặt *Adapter Pattern* dựa trên ý tưởng về inheritance. Lớp Adapter của chúng ta sẽ kế thừa lớp Adaptee và thực thi cả giao diện mới (giao diện mà Client cần). Các phương thức trong giao diện mới sẽ được cài đặt bằng cách gọi lại các phương thức tương ứng của Adaptee mà nó kế thừa được.

*Class Adapter* tương tự với trường hợp sử dụng thiết bị output trên PC của chúng ta. Hãy tưởng tượng bạn có một PC và một màn hình, người dùng là Client, màn hình hiển thị của PC là Adaptee (coi nó là một “giao diện” hiển thị), và máy in là TargetInterface. Giả sử giờ bạn không cần hiển thị một văn bản trên màn hình nữa, mà bạn muốn nó “hiển thị” ra giấy; do đó, bạn cần đến giao diện của máy in (TargetInterface). Vì vậy, bạn sẽ cắm máy in vào máy tính của bạn, và giờ đây nó vừa có thể hiển thị lên màn hình, vừa có thể “hiển thị” ra giấy, lúc này hình ảnh của PC được cắm thêm máy in vào tương tự như một *Class Adapter*. Bộ PC này (PC + màn hình + máy in) sẽ có cả giao diện hiển thị của màn hình và giao diện hiển thị của máy tin. Tương tự, trong *Class Adapter Pattern*, Adapter sẽ kế thừa cả giao diện của Adaptee và TargetInterface.



**Vấn đề**  
  
Ta viết phần mềm hiển thị dữ liệu (giá vàng, cổ phiếu, biểu đồ…) trên các thiết bị, môi trường (window app, web, mobile,tablet….. ).  
  
Khi có sự thay đổi số liệu ở cơ sở dữ liệu (CSDL), các thiết bị sẽ lần lượt cập nhật và hiển thị lại dữ liệu ngay lập tức.  
  
Thêm một thiết bị mới mà không cần phải sửa mã nguồn hay thay đổi cấu trúc.  
  
**Observer Pattern là gì**  
  
Xác định một sự phụ thuộc một-nhiều giữa các đối tượng để khi một đối tượng thay đổi trạng thái, tất cả các đối tượng phụ thuộc nó được thông báo và cập nhật tự động.  
  
Sự thay đổi của một đối tượng ảnh hưởng đến sự thay đổi của những đối tượng khác và chúng ta không biết có bao nhiêu đối tượng cần phải cập nhật.  
  
Một đối tượng có thể gửi thông điệp đến những đối tượng khác mà không cần biết chúng là ai.  
  
[![Click image for larger version. 

Name: Observer_01.jpg 
Views: 25 
Size: 27.5 KB 
ID: 411](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDgRXhpZgAATU0AKgAAAAgABAE7AAIAAAAHAAAISodpAAQAAAABAAAIUpydAAEAAAAOAAAQyuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEh1bmdYdQAAAAWQAwACAAAAFAAAEKCQBAACAAAAFAAAELSSkQACAAAAAzY0AACSkgACAAAAAzY0AADqHAAHAAAIDAAACJQAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDEyOjA3OjA0IDE1OjAwOjQ0ADIwMTI6MDc6MDQgMTU6MDA6NDQAAABIAHUAbgBnAFgAdQAAAP/hCxlodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDEyLTA3LTA0VDE1OjAwOjQ0LjY0MjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5IdW5nWHU8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgA/QJWAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAp6hrGmaSIzquo2lkJc7PtM6x78dcbiM9R+dUv+Ex8Mf9DHpP8A4HRf/FVS1SCKf4k6Ek8SSKNL1A4dQRnzbT1rZntdLtYGmube0iiQZZ3RQAPrQBT/AOEx8Mf9DHpP/gdF/wDFUf8ACY+GP+hj0n/wOi/+Kqex/sTVLcz6ctjdRBipeFUYZHbIqU2uli4WA29oJmUsI9i7iB1OKAKf/CY+GP8AoY9J/wDA6L/4qj/hMfDH/Qx6T/4HRf8AxVWo7fSZbma3igs3mgx5saopZMjIyO2RU39m2P8Az5W//fpf8KAM/wD4THwx/wBDHpP/AIHRf/FUf8Jj4Y/6GPSf/A6L/wCKq8thp7Z22lscHBxGvH6U7+zbH/nyt/8Av0v+FAGf/wAJj4Y/6GPSf/A6L/4qj/hMfDH/AEMek/8AgdF/8VWh/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBn/8ACY+GP+hj0n/wOi/+Ko/4THwx/wBDHpP/AIHRf/FVof2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAZ48Y+GScDxFpP8A4HRf/FVs1ynxA0+zT4aeJmS0gVl0m6IIjAIPlN7V1dABRTI5UlBMTq4VipKnOCOCKfQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUVyNjd+K9an1CWw1PRrO2t76a2jjm0uWZ8RttyWFwoJP+6KufYvGv/Qe0H/wSTf/ACXQB0VFc79i8a/9B7Qf/BJN/wDJdH2Lxr/0HtB/8Ek3/wAl0AdFRXO/YvGv/Qe0H/wSTf8AyXR9i8a/9B7Qf/BJN/8AJdAHRUVzv2Lxr/0HtB/8Ek3/AMl0fYvGv/Qe0H/wSTf/ACXQB0VFZHhXVLnWfC1jqF+IhcTITJ5KlUyGI4BJIHHqa16ACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDnb/wD5KXoX/YK1D/0baVj/ABXgS98M21pKb2KOS7RzcWtt9oEJXkF0wdyk8dK1dTljh+JOhNLIqL/ZeoDLHA/1tpW99utP+fqH/v4KAPGtF1TXdBfRZrnS57W2nur+4u206wKfbisYWEvGAdhbOe33RWLMvie5uBcag+tR+IV0iGOzlghIVp5ZDIwZsbQqgqCDjpXv/wButP8An6h/7+CmyXVjNE8cs8Do6lWUuMEHqKAPC7vX/F11FHcJc3j6df3d2zT2kZZ28oLHFEGRSQCQzDPU8Zra0qbxLPrqWHiC/wBdW9sltkha0g2xTJ5YaSRyRgktlSM8Yr1Wy/srTbKKz09rW2toV2xxRMqqo9ABU/260/5+of8Av4KAOO+E2mz2Pgx5r5rz7Xe3k1xOt2MMjFyMAYHHAruKg+3Wn/P1D/38FH260/5+of8Av4KAJ6Kg+3Wn/P1D/wB/BR9utP8An6h/7+CgCeioPt1p/wA/UP8A38FH260/5+of+/goAxPiH/yTLxP/ANgi6/8ARLVQ8U3V/Hq8kSTXkKGzzZfZoyQ9xu6MQD7dcDBNWPiFd2z/AAz8TItzES2k3QAEgyf3LVP/AMITYf8AQS17/wAHd1/8coA4nTzrFr50Vo2pW+pyXt8bpfJLRJCY5WR0GME+Z5eMHJyc8Uwaj4haxgjVrlbNbn/SLk+cVceWSNvy+YBvAyMdT6V3P/CE2H/QS17/AMHd1/8AHKP+EJsP+glr3/g7uv8A45QBzNtbeJ7i1u72W+vGubK0t5reNV2JcOp3uhUjOXChT6Zqrqd94jmSC5aa5tYryOW5hUxyExsXAiUhATkIASDxktXYf8ITYf8AQS17/wAHd1/8co/4Qmw/6CWvf+Du6/8AjlAHLX51y3s5rue+vfMe/wBnluj+UY1RiF+QFlBbnPPIAPFd3oNxNdeHrCe6imimeBC6TkFwcfxEAcnr0FZv/CE2H/QS17/wd3X/AMco/wCEJsP+glr3/g7uv/jlAHRUVzv/AAhNh/0Ete/8Hd1/8co/4Qmw/wCglr3/AIO7r/45QB0VFc7/AMITYf8AQS17/wAHd1/8co/4Qmw/6CWvf+Du6/8AjlAHRUVzv/CE2H/QS17/AMHd1/8AHKP+EJsP+glr3/g7uv8A45QB0VFc7/whNh/0Ete/8Hd1/wDHKP8AhCbD/oJa9/4O7r/45QB0VFc7/wAITYf9BLXv/B3df/HKP+EJsP8AoJa9/wCDu6/+OUAdFRXO/wDCE2H/AEEte/8AB3df/HKP+EJsP+glr3/g7uv/AI5QB0VFc7/whNh/0Ete/wDB3df/AByj/hCbD/oJa9/4O7r/AOOUAdFRXO/8ITYf9BLXv/B3df8Axyj/AIQmw/6CWvf+Du6/+OUAdFRXO/8ACE2H/QS17/wd3X/xyj/hCbD/AKCWvf8Ag7uv/jlADtL8e+FdZneDTdfsZZo5/s5iMwRzJnG0K2CTnjijX/Hfhfwtf29l4g1q1sbm4GYopWOSM4zwOBnua8LsP2Vr+61ia91zxNDaxm8Mix2UbTO0W7P+sfbtfrztbnnnpUnin9la4uLyCXwx4kaSM4WZdXyzqOcsrovPbClR3+agD2zwU6yWWrPGwZW1e7KspyCPMPNYmseJ9fvdZ1+38PT2VlZ+H4VNxPcx+YZpSnmbAMjAC4yfU1Y+E+hx+GfB02iwzyXCWOo3MIll+8+JDyat33w60i/1e+vWuL+GPUirX1nDclIbllAALKOegAOCAe9AHJaZ8X70mS+1XSJRpkFpZPcSQKD5Mk55JJPIG5OBzzW6vxb0M3upJJFLHa6aJBNcMyDDIcEbM7hk8A4watSfC/QZNXlvHe88qa4juZbEXBFvJJHjYSnouBgdOBVVfg/4aMM6XTX128sZijluLgu8K79+Fb6gcnJ4oAhtfi/pd5Eq2unXdxdyXos4rWBkkaRjH5mQQcYC9eeKsn4raPFFKbq3nikt5LhLmPg+SICA7MemNxAHrmtPT/Aml2F5pt4Zru5udOeWSOWeXcWaRQpJ4xwBgYwBk1BL8NPDsw1zfBKTrkgluWMpJRhz8n935ufrQBk2vxh0m9gxbWF1PeNdraRWsDJI0rNGXBBBxjA554r0CF2lt45HjMTMoYo3VSR0Nc1pvgDSdNutPuvMurq4sJZJopbibcS7rsJI6cLwMAYya6igDnfAP/Ii6Z/uN/6G1dFXO+Af+RF0z/cb/wBDauioAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOT1/S7DVviHoMGq2Ntewrpt+4juYVkUN5loM4YEZ5PPvWj/wAIT4V/6FnR/wDwAi/+JqC//wCSl6F/2CtQ/wDRtpVH4hJd6pp9v4f02OZ5dQYmUwuEZYlGSckgDJx3oA1f+EJ8K/8AQs6P/wCAEX/xNH/CE+Ff+hZ0f/wAi/8Aia8z1u/1S90ZpJtQurG4h8PXUM0S7P8AXwyIHxlTnhTkjtzWpqPjXVbbXZItNvTdIrTQhJDHuZkiypESoT1BOSRnPAxigDuP+EJ8K/8AQs6P/wCAEX/xNJ/whfhQEA+GtGyen+gRf/E1yn2m61fVNEW18STXQN2XeWK3QGAm0lJTft2nnBwRkZA9KpSX+qavpV9Jezyvcw2UYJiUBniM7CVgFGPuouSM8E9KAO1j8I+D5VdovD+huEJDFbKE7SOoPy8VJ/whPhX/AKFnR/8AwAi/+JrhPFz6a1vK/hCWERRaVP8AbZLVd6omU8rOM/NkMRnnAbp1qfVfE2p2zziw19bqwglB+0GSKOSX92SyI/l+WxUgHacZ3Yz2oA7T/hCfCv8A0LOj/wDgBF/8TR/whPhX/oWdH/8AACL/AOJrkbfxXrlz4iWI3McCm6hiitZmVXkiKBnYx+WWyQSwIYKMAdM5LHW/E8Oj6PPPfPdza7boIibdALeckHsowmzd1zzQB13/AAhPhX/oWdH/APACL/4mmjwZ4TJIHhvRiR1AsIuP/Ha4m98W+JYLTWLmG6gE1rZ38v2ZmVmi8pD5bCMRgjDgZ3MQd3Tpjr9OjmsvGJtDcyXW/T/MuZpAAWkEgCkhQAOC3T0oAyfHnhDw1a/DrxHPbeHtKhmi0u5eOSOyjVkYRMQQQuQQe9d1XO/EP/kmXif/ALBF1/6JauioAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOE8PHxKH1n+yV0o239r3W37U0gfPmHOdoxWznxr/c0H/vub/Cjwb/x7av/ANhi7/8ARhp83iN4rnXz5SG20iBSG53PKULsp9gNnv8AN9MgDM+Nf7mg/wDfc3+FGfGv9zQf++5v8KrWvj2yNpby39vPCrkRz3KpmCGTyy5UknccBTkgEDuRSr413619kGl3iRCzFz+8jAd97hYwBuwufmJ3FSMcgUAWM+Nf7mg/99zf4UZ8a/3NB/77m/wqq3j+xF5aL9nmW2lhnlnmcD9z5bKoGFJ3FmJA25ycYzmp/wDhL0upEtNPtJV1B7lbfyLjb8mU3lyUYjATJxnPGKAH58a/3NB/77m/woz41/uaD/33N/hWhoepSanZzPOqrJDcSQNsBAJRsZGea0qAOb+Hu/8A4QDSvN27/Lbdt6Z3tnFdJXO+Af8AkRdM/wBxv/Q2roqACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDnb/wD5KXoX/YK1D/0baVpanrVrpM1tFcLK8t0zLFHEm4sVGT+lZt//AMlL0L/sFah/6NtKTxFb6gviDRNSsdPmv4bNpvOigkiVxvj2gjzGUHn3/CgDUttT07UbQXMc0TRhzETJgFXzgqQeh7YpwvNN8+aHzrcSW5XzVJA2Ej5c++K4K98Ka1LdDUpba4dbiS7kexs2t2eJpdmzcZTsOFQ5K5IJ4yCaH8Laxa3WyHTZpEcWplukkgllJjj2nHmMAzZxlmUA9QM0AeiB7e32IrRR+YSUUEDeepwO/rWfDqGj3UX9txTxERQndLnDCPPQjr1FcVo/hrWbS0t4dV0aW+ka2t4Eka6jX7IUkJZmIbOOQw2Ak9DtFNn8HX+neHYBDbJE8Vo63RiYHcfPWTnB54B/+tQB2EvinSrSSWOOOZ1hUSXLwQFltwRuBfHTjnAycVbsdW0rUXe3s5oZPKK4UYwcqGBX1GCDkVg2sWpaVc6s2m6U2pQatL9qt545o0WNmQDbKGYMACOqBuO2Rzz2leC9Y0kixGnCaVZLQR6qkyIFWKJVYn5t45BAwDwefcA9L8+1Jkl82HMIIkfcPkA6gntSm5gVFZp4wr/dJcYb6V5fD4T8RpBLCtpcC0jnt5XSRrY3E6o7HYrA7HUEqwMoUnBHoRo2HgW4uJYhfxTpALC5SM3MsbPbyySqy8R4UcAnjIGcZoA7+SSCAPLK8cY4DOxA9hk/jVS5msNLmlvbhlSSbYkjk8hQcDPsCT+dcK/hPX5IrbUNQFzPcPJMbm0tWgZuUWOJh5p8v5VQ575fI7gyr4cuNN0jUJtUhMssunWtpaiWRZXRxvBTgAHBZeQBnHtQB0XxC5+GPicj/oEXX/olq6OuY8cxvF8JvEMchy6aLcqx9SIGzXT0AIzBVLHoBmuXs/HUWoWMF5Z6Brs1vcRrLFItmMOjDII+buCK6ab/AFD/AO6f5VheBGCfDXw2zEBRpFqST2/crQA3/hL2/wChb17/AMAx/wDFUf8ACXt/0Levf+AY/wDiqdp/jbQr+xS6fULa1jlmeKLz7iMGQq23jDHv261rw6lZXF5LaQXcElzCMywpICyD3HUUAY3/AAl7f9C3r3/gGP8A4qj/AIS9v+hb17/wDH/xVWLjxbo0UDSW97FelLmC1dLWRXaN5ZViXdzwNzc/Q4yeKntPEGn3OgR6zJMLSzkUtvuWVNvJHJzjqPWgCh/wl7f9C3r3/gGP/iqP+Evb/oW9e/8AAMf/ABVakmsWY0dtTt5o7q1Vd4khlTaw9QzMF/M0W+uaVd3zWVrqVrNdJu3QRzKzjaSDwDnggg/SgDL/AOEvb/oW9e/8Ax/8VR/wl7f9C3r3/gGP/iq1ZNa02PUGsGvrf7aql/swkBkxjP3evSqGl+MtA1bRI9Ut9TtkgaCOeQSToGgDgEB8EgHnHXrQBD/wl7f9C3r3/gGP/iqP+Evb/oW9e/8AAMf/ABVaDeI9ERLZ31eyVbv/AI9ybhcS84+Xnnn0qSLWtLn1B7CHUbWS8TO63WZS64znK5z2P5UAZf8Awl7f9C3r3/gGP/iqP+Evb/oW9e/8Ax/8VWl/wkGj/bYrP+1LP7TMMxw+eu9/oM5PSoR4r8PsyKut6eS77EAuU+ZuOBz15H5igDOufGy2lrLcT+HdfWKFDI5FkDhQMngNk/hWXZ/FzQ7+ziu7LT9XngmUMkiWgIYf99V193qUVpf2do0ckkl4zBNgGFCjJJJI46dMnnp1x5bqGg6Bomr63AdWs/D10lyLu3eWREhuEmBYRybuSfMSU5HKqygHHygA6W4+LOi2lrLc3Om6zHDChkkc2fCqBkn73pXc185654k06+8OeINNW9tHvodOnZ0tpxNGw8s/NHIAA46ehGRkA19GUAFFFFABRRRQAUUUUAFFFFAHO+Df+PbV/wDsMXf/AKMNNufB63F3qRN/Itlqkyz3dt5YJdlREwH6gEIuRiqtja+K9Fn1CKx0rR7y3uL6a5jlm1WWF9sjbsFRbsAR/vGrn27xp/0L+hf+Dyb/AORKAMq1+H5vNGlsvEV41xDM08n2ZVG2KSVWQsG/iwrkDPTPfjFm78BrqF697f6m9zcSGPzBLArRMqK4C+WeP4y3OeeaufbvGn/Qv6F/4PJv/kSj7d40/wChf0L/AMHk3/yJQBnD4b2P9nJZNdM8KW0UAVoUIJjn85WI6HLYBGOQO1WbbwkdIubO50tLZp4vOMhEKwqWdcAhVGBjAH0zzUWkeIvFutaTBqFr4c0ZIZwSok1uUMMEjnFqfSrv27xp/wBC/oX/AIPJv/kSgDU0bS49H0uK0jYuVyzuersTkn8zV6ud+3eNP+hf0L/weTf/ACJR9u8af9C/oX/g8m/+RKADwD/yIumf7jf+htXRVj+FNMutG8LWNhf+V9phQiTyXLICWJ4JAJHPXArYoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOdv8A/kpehf8AYK1D/wBG2lS+I/FEXhy80uO4t3khvpmieVekAC53n2qK/wD+Sl6F/wBgrUP/AEbaVf1PRIdU1GyubhspaiVWhIyJBIm0g+lAFH/hMtPikmS73oyTyxIsamRnWPG58AdBkfmKq3Hj7T4ri8iiUv8AZVVwxziZWjMg2cc8D9ar6b4Eu9Fcy6ZrAab/AEhA13b+YPLlZWAwGU7l2AZzyM5HTDU+HixssMeoYtIoIookMPzjZEYsls4PBz0FAGppPjDT7/R4bu5nSKVpIoXjAb5ZJACi8jPIYVLpXiRdTlssw+XFqEDXFq+c70B4z6EqQ341jr4RlbxpZTMsn2CysQHZtoSecK0aMADnKpJIDn1THSrejaJewT6HFeQCOLRrNoEkDgiRsCMEAHoUGeehOKAOpqK3ure7V2tZ4phG5jcxuG2sOqnHQj0rP8T6bc6v4avLCzdVkuECMrMVEse4b49w5Tem5N4BK7twBIxWLoOn3p8TJfW+kXXh7TYbMwSWUzW5+0vkbGxE7jCjcMkg89xmgDr6KxrfR9ThkuGl8S304liZI1eC3AhY9HXEYyR2DZHqDT7bSdRgsLmCbxDe3M0oHl3LwQBoP90LGFP/AAIGgDWorJi0nUY9JntZPEN7LcSMCl40EAkiHHAURhCDg9VJ5PtRPpOoy6Zb20XiG9hniJMl2kEBebrwQYyo/ADpQBT+If8AyTLxP/2CLr/0S1dFXF/EXTr5vhfrAXW7pTbaVdGdhFD/AKWPKPD/ACfL0P3NvX6V0F9peoXWoJcWuvXllCoG62ihgZGwecl0Lc9ODQBozf6h/wDdP8qw/AX/ACTfw1/2CbX/ANErVi+0vUJtQN3Dr15b2y4JskhgMbAdQWaMvz3w3fjFV/AX/JN/DX/YJtf/AEStAGBN4L1VbG7toGtZft+nPp8jyk/uMvIwcDv/AKzpxyo/DoNE0G40i01aPz0ea9uXmSUrzzGijd6nK1vVy/izUZbS9soZdRk0uxkSRnuk28yBflTJHfk44zjFAGDa+Btae+E17PEQBZHJfPMF0szYAACgqCAB+Nbi+G7q18JaZZiSF7jTp1uMOcJIQxbBPbr19an0bU/EFxo1jLc6UkjyQo0kj3Hltk4ydmzg9eKytTvtdvdL1iytbae+tPs8wW8e3NtIrjPyBW5kz2ZVA92oAuTeHLqfwfqel+fBFe6m8s6qOUTc4YgDvjI59TTtN8K3FjeabM0sR+y317cybRy6ztIVH1G8flXJ+JfEBeLTdR8PXw8zTdHmE88a7xbPJJbKoYYI3fK/ynng8Hobx1y6+3SWw16YaJ5pEWrAo2X8rds34wRnGPUnHtQBrv4YvrjUZU32z2MuprqInyTIAB9wfU/Ln+6WFZkHgbUjpGnW91Haq+l6fFZRCKRl83YUO8MPun5OM55Jrn9L8TapY6Tolvb3bR+TLYRSieURtJHLMglZYthLjYzZYlQuDjpTrbU9c02witLXVwsaS3zia7ufKzKLlgsfELliFIOzAJB4HoAbMvgXVZXkF1qSzve2n2OYGUIUXe7AnA+b5WxxgkjP017zwbdy6XcJa3EUV4+oy3UcnOAknyspx1JXI/KqSyXD6Xc310oXU1120VuSMAzxRjqMgGNjwR0Y139AHCDwNdQa3iGUyaebq3uRumK+X5IXA2gcnKDvjHX3rzeBNSh0jR7axkhS5s7cxPcRtsGS4c7kIIdT1IIznvXodFAGDqUMsfibQLkqzoizwOyKdql1Ugn0/wBWR+Na4srVb9r5baEXbxiJrgRjzCgOQpbrjPOKnooA5L4naXp+ofDfxBNf2Ntcy2ulXclu80Ku0LeSx3KSPlPA5HoK62ud+If/ACTDxT/2B7v/ANEvXRUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVheKrqU2cOk2TbbzVZPs6N/cTGZH9OEB4yCe3NAFf4eyxy+BNOMUivtDq205wQ7ZB966WuX061i8K+JV0y3TytM1NN9sO0dwg+dM9Muo3gcklZCeBXUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAc3r1vqsPirStW0vTf7Qjt7S6t5YxOsRUyPAyn5uo/dN+lO/tvxH/0KUn/gwhroqKAOd/tvxH/0KUn/AIMIaP7b8R/9ClJ/4MIa6KigDnf7b8R/9ClJ/wCDCGj+2/Ef/QpSf+DCGuiooA5K88W63Y3Vhb3HhOUSX85t4MX8Ry4jeTn0+WNv0q3/AG34j/6FKT/wYQ0nicga/wCEMnH/ABOH/wDSG6ro6AOd/tvxH/0KUn/gwho/tvxH/wBClJ/4MIa6KigDnf7b8R/9ClJ/4MIaP7b8R/8AQpSf+DCGuiooA4nxLN4n13wnq2kQ+FjFJf2U1ssj6hEQhdCoJx2Ga7aiigBk3+of/dP8qw/AX/JN/DX/AGCbX/0Stbk3+of/AHT/ACrC8Bf8k38Nf9gm1/8ARK0AXNCvpr22uVuZFkltrqWAsoxkKxAOPXGK1KzdD0x9KspYpZEkkmuJZ3ZF2jLsWxjJ9a0qACiiigCC+sbbUrN7S9iEsEmN6EkA4II6e4FT0UUAFFFFAFRtKsnvftbW6mfzFl3ZP3wpQNjpnaSP/wBVW6KKACiiigAooooA534h/wDJMPFP/YHu/wD0S9dFXO/EP/kmHin/ALA93/6JeuioAKKKKACiiigAooooAKKKKACiiigArjW1O4s/Gup3l/omq3HlItrZPa23mJ5RVXdgcjlnOCD08sY6muyooA4LVvE0fifw35mkaPrXngi4sLr7CdqTIcqTznaSNrAdVLDvXa6fcyXmm21zPbSWks0KSPby43xEgEo2OMjOD9KxfAP/ACImmf7jf+htXRUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVmeJL6603w3e3enwS3FzHH+7SGIyPknG4IAS2M5wBzjFAHP6lpQ8banfkyGO30v9xp8oG7y71WDm4APBMbKijPcSA5DV0OgaqdY0aG6ljENyMxXMAOfKmU7XX3GQcHuMHvXO6Z4v8PaBplhpvla2hP7qPfoF+Gmk2l2xmHJJCs2OTgE9qdo2sQyeN510yz1UWepReZObnSbm2SKdBjduljVfmXAPJJKrxigDsqKKKACiiigAooooAR13Iy9MjFcnpXhnxLo+jWWmWnie1NvZW6W8W/SsttRQoyfN64FdbRQBzv9leK/+hns/wDwU/8A22j+yvFf/Qz2f/gp/wDttdFRQBzv9leK/wDoZ7P/AMFP/wBto/srxX/0M9n/AOCn/wC210VFAHO/2V4r/wChns//AAU//baP7K8V/wDQz2f/AIKf/ttdFRQBzv8AZXiv/oZ7P/wU/wD22j+yvFf/AEM9n/4Kf/ttdFRQBzv9leK/+hns/wDwU/8A22j+yvFf/Qz2f/gp/wDttdFRQBzv9leK/wDoZ7P/AMFP/wBto/srxX/0M9n/AOCn/wC210VFAHO/2V4r/wChns//AAU//baP7K8V/wDQz2f/AIKf/ttdFRQByOreGPEmtaLfaXe+J7b7PfW8ltLs0rDbHUqcHzeDgmuuoooAKKKKACiiigArmtZutZn8W2mk6RqFvYxvYy3Mjy2vnFiskagD5lx9810tc7N/yU+z/wCwPcf+joaAD+yvFf8A0M9n/wCCn/7bR/ZXiv8A6Gez/wDBT/8AbaqeKfEGqW/ibRvDmgG2hu9SWWaS6uULiGOMDouRuYkgdeBn8K0nja/0CGHT/EGnHUdaImlaPSSpU28Z/wBcwkZdvBHygk5zj2ANT+yvFf8A0M9n/wCCn/7bR/ZXiv8A6Gez/wDBT/8Abax5vi1oiXDC3stSurWL7Kbi9hiTyoVuCBGxy4Y8sAQqkj06ZmPxM09U1C6/snVm0yy84f2jHArwytCSJApDZHIwCwAbBxQBpf2V4r/6Gez/APBT/wDbaP7K8V/9DPZ/+Cn/AO21z918Y9FsWhN9p2o2yNFHPMZhEpt4pG2xuyl8nPXChiByarW/xjgRL0alod7HNDqs9hBHGYx5qxDLOS7gAgdRnuMZ5wAbuleF/Emj6XDYWnie2MMIIXfpWTySeT5vvVz+yvFf/Qz2f/gp/wDttc9D4z1bxjrdxaeB7qztILOyiumlvrdnMzSglEwCNqjHJ5NZOi/GiW+WO+udFnk00aVDe3L2YVjas0kkblizDKgpwFBODnpQB2/9leK/+hns/wDwU/8A22j+yvFf/Qz2f/gp/wDttY8HxZ0O41q/slgu1t9PjkkuLwhNiCNN7ZQNvA25wSoBxxVWH4zaLNavIulao02+FYbaNYpHmE2fLZdshHUHIJyO460AdF/ZXiv/AKGez/8ABT/9to/srxX/ANDPZ/8Agp/+21QPxN0aGYwXttf2txHPLFPDJEpaARRea8jbWPyBccjJOQADms63+Mmhy2ZlfT9SScmHyrRViklkSY4RhtkKjnqCcjIyKAOg/srxX/0M9n/4Kf8A7bVYT+IdK8TaPa6jq1rfW1/JLG6JY+Sy7YmcENvPdfSuohkMsCSNE8JdQTHJjcvscEjP0NYGvf8AI3+Fv+vq4/8ASaSgDoqKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKwtb1zUbLWrDS9H023vri8gnuCbi7NuqLE0SnkRuSSZR2HQ1D/aHjH/oXNI/8HUn/AMjUAdHRXOf2h4x/6FzSP/B1J/8AI1H9oeMf+hc0j/wdSf8AyNQB0dFc5/aHjH/oXNI/8HUn/wAjUf2h4x/6FzSP/B1J/wDI1AC+Jv8AkP8AhD/sMP8A+kN3XRVxeqReMtR1DR7kaFo8f9m3rXRX+2JD5mYJYtv/AB78f63Oefu471o/2h4x/wChc0j/AMHUn/yNQB0dFc5/aHjH/oXNI/8AB1J/8jUf2h4x/wChc0j/AMHUn/yNQB0dFc5/aHjH/oXNI/8AB1J/8jUf2h4x/wChc0j/AMHUn/yNQB0dFchq/iTxVomiX2q3nhvS2t7G3kuJRHrDliqKWOAbcAnA9RXX0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFUJ9c0621JLCa6VbhyBt2khSfuhmxhSewJBPbNZ+qeM9I0ywtrs3KTR3KJLFsP3o2ZV3j1GWXjrzQBv0Vi/8ACWaSSji7iWAxzO7yEo0fllAwKMA2cuP09aV/FuiR2v2h73C7zGU8l/MVgoYgx7dwwpDHI6EHpzQBs1zs3/JT7P8A7A9x/wCjoau6P4isddnvY9PZnFnII2fHyvlQcqe45rD1ufVIPiRYNo1ja3sp0m4Dpc3bQBR50PIIjfJ9sCgDV8Q+FbTxDc2N29zdWV9p7O1rd2jhXj3DDD5gQQR2IrIf4ZaWLaJLLUNStLhRKJbuOVTLciUgyeYWUg5wOgGMDFaH9oeMf+hc0j/wdSf/ACNR/aHjH/oXNI/8HUn/AMjUAUn+GHh5rK8tI1uYobsWodEkHAtyDHjIOOVGfXmmH4Y6Uy38L6hqbWV4ZmFn548qEzHMhUbc5JJxknGTitD+0PGP/QuaR/4OpP8A5Go/tDxj/wBC5pH/AIOpP/kagClqfw00bU9VgvzLdW8iQRW8oiKf6RHGcoGJUkdOqkEiorv4V6Hcz3E8c95BNPeyXodDG3lSSDEm0MhGGHXOegxitL+0PGP/AELmkf8Ag6k/+RqP7Q8Y/wDQuaR/4OpP/kagCpc/DrT3mWbTdR1HS5zarZzy2kqhriIdA+5SMjJwRgjNVbr4T6DMBHZ3F/p9sbOKxktbaYCOSCNiwUhlJ5JOSCD+tav9oeMf+hc0j/wdSf8AyNR/aHjH/oXNI/8AB1J/8jUAZ1z8K/D17rX2++NzdKElRLaZ1ZIxJGY2AO3fjaSAC2B2FJp/wu0exWHfd3ty1vPDNC0jRrs8rhFwqAEepxk4HNaX9oeMf+hc0j/wdSf/ACNR/aHjH/oXNI/8HUn/AMjUANufAOgXniS81u5tTJc3to9pMhb92UcBWO3H3ioAJ9BWfY/C3RbOCGN7i7n+zyxSQM/lgxiI5RflQAj1JyT61pf2h4x/6FzSP/B1J/8AI1H9oeMf+hc0j/wdSf8AyNQB0dc7r3/I3+Fv+vq4/wDSaSk/tDxj/wBC5pH/AIOpP/kasu6utdn8beGV1nS7GyiE9wUe2v2nLN9nfggxJge+T9KAO3rL8Qa02h2EU8dm97LNcR28cKOqFmdto5bgCtSud8Zf8e2kf9hi0/8ARgoAP7c8Rf8AQozf+B8H+NH9ueIv+hRm/wDA+D/GuioyM4oA53+3PEX/AEKM3/gfB/jR/bniL/oUZv8AwPg/xroqKAOd/tzxF/0KM3/gfB/jR/bniL/oUZv/AAPg/wAa6KigDnf7c8Rf9CjN/wCB8H+NH9ueIv8AoUZv/A+D/Gt9Z4ndkSVGdfvKGBI+tPoA53+3PEX/AEKM3/gfB/jR/bniL/oUZv8AwPg/xroqKAOd/tzxF/0KM3/gfB/jR/bniL/oUZv/AAPg/wAa6KigDnf7c8Rf9CjN/wCB8H+NZeu+PNT8OWcd3qvhO6jgeQReYt5CwVj0zg8ZPH1NdtVPVtOi1bSbmxnVSs0ZUFlyFOOGx7HmgDi1+J10yhl8LXRUjIIvIcH9auaP8Qn1PxDZaVc6Fc2RvC4jleeN1BVC5BCnPRTXDRjTDodvdeGZriW+RPLu9PtrO4uoPNUlWLvBFIYZDjdtPr053VH4G8TaX4n8aeHbnSbjzNk0wlidSrxE2znDA/zGRwcE4oA9xooooAKKKKACiiigDnb/AP5KXoX/AGCtQ/8ARtpUfivxHfaZqOlaPotvDLqOqSMqPcEiOJVGWZscn0wKkv8A/kpehf8AYK1D/wBG2lTeIvDMevS2N1HeT2F9p8pltrmAKSpIwQQwIII7EUAZw8Y3Gj2tvbeKdPcatNJIsdvpymfzo0wWlUdQuCOvc4qCb4reGY/LMcl3PG1vFdPJDbMyxRSMVDMe3zAg+lEnw7INrd22v38WrQGfdqDKjtKJsb1ZSu3HyrjAGMVXj+Eujw6bc2MF5dpFcWENkxyrELG5fdyPvFmOe1AGmPiJobXVzGv2toLZZi12tuxhYxKWkUN3IAPHtWc/xe8MxwW8zm8WKePzwzW5G2EkAStnopJ4NLa/DNLVL20j1/URpdytxixGzYrTKVZs7ctjJIB4BNM1T4T6XqL6c6XtxA9nYR6ezbEczQpjGdynB46jHU0AR2/xf0aT7cJrO9RoNQNjAqwljcNtL5X6KCT6cetZmufFmSPVbYaK2nx6fPaLcQXGos8a3TFyrRK2MKw2nOa1bz4TadczTzR6jdRyPfG/hBSN1gkMZRsArghgeQc9Bin3fwwhn0+OxttavLe1Nr9mni8uJ1lGSSwDKQjHJ5XFAGs3jzRoopzPK4e2ube0mCIWAlm27AD3GXHNc7pvxbs7zSZZL+3lsLjddLDK8LNBIYWYYDdztUHFWLj4UWcl5m21i+trNprW4e1QIwaS327CWKk4+UZFUdR+Ezx6HNbafqt1feSty9lZ3TIkSSz5yxZUzgbj60AbNv8AFDw8dRstNnuna4uDFG0yREQrLIu5ELHoSOxqtqXxa0i08PT6tZ2OoXMSbGiP2ZlWdGcLvQ9xUFj8I7GDVbe/mvpeGhmuLVY0KSTRoFyGI3AHaDgHqKsQfC6KPQJNFm8Q6pPYiFYLWJimLZVYMuML8xGAMtngUAa3juYXHwp8RzKrKJNGuWCuMEZhbqK3rzULPT0V7+7gtUY4VppAgJ9Bk1gePYmi+FHiSJpGdl0a6UyHAJ/ctzxXh/iz9mjxXqFxJPYeLo9VMtw8hTUi6FQSTncNwLc84Uf0oA+lIZoriFJreRJYnGVdGDKw9QR1p9ef+CPB3iXR/A2kadd+JbuxntrVI3to7e2kWIgdAxQkj3Jre/4R/Xv+hyv/APwDtf8A43QB0VFc7/wj+vf9Dlf/APgHa/8Axuj/AIR/Xv8Aocr/AP8AAO1/+N0AdFRXO/8ACP69/wBDlf8A/gHa/wDxuj/hH9e/6HK//wDAO1/+N0AdFRXO/wDCP69/0OV//wCAdr/8bo/4R/Xv+hyv/wDwDtf/AI3QB0VFc7/wj+vf9Dlf/wDgHa//ABuj/hH9e/6HK/8A/AO1/wDjdAHRUVzv/CP69/0OV/8A+Adr/wDG6P8AhH9e/wChyv8A/wAA7X/43QB0VFc7/wAI/r3/AEOV/wD+Adr/APG6P+Ef17/ocr//AMA7X/43QB0VFc7/AMI/r3/Q5X//AIB2v/xuj/hH9e/6HK//APAO1/8AjdAHRUVzv/CP69/0OV//AOAdr/8AG6P+Ef17/ocr/wD8A7X/AON0AQ6t4Wur7xTb6raXKWxR4y8sbukhRWBZGUHZIGGR8wyueO1Y6eANUFvp0bX9mv8AZdmlnbARsQyo6MGbngkRgYHA/St7/hH9e/6HK/8A/AO1/wDjdH/CP69/0OV//wCAdr/8boA57VfBF9q2q7LiYpJPFNNNcw7lSNzNA8cYIIJ/1JyeuD71o2fg++sJxf2stql8XkBEryzr5bKFHzyMWJGM9hzj3rQ/4R/Xv+hyv/8AwDtf/jdH/CP69/0OV/8A+Adr/wDG6ALWh6RcaVdX7TzRypcyJIpUFSCECnI6AZHGKqzf8lPs/wDsD3H/AKOho/4R/Xv+hyv/APwDtf8A43WfYWF9Y/Ey3GoavPqZfSJ9pmhjj2fvos42KM59/SgDdvNWe28V6ZpQjUx3ttcys/dTGYgB+PmH8hWTa+ObdpruG7tZ/MgnuFxBGXAihZVZ2Pb7wrV1bQxqWoWOoQXT2t5Y+YIpFUMCr43KQex2r+Vc7/wgVzFqi/ZdUmjtpbS4jvJyEMk8k0is3BXCggH7uMYAoA0J/G9o17p8WnRS3MN1efZWuQh8tT5bOcN6/KP1qKb4hWAiQ21tcyymaBGhMeHCSnAbH9KRvh/ABHBb6pd29jHM1wttGEwJGjZCd2N38RPXrUNt8OYLYqy6lNuRYQpEUa4MT7lJwOSehzQBbufiDo0NvdvE7yPBbyzxLjAn8vghT65IH41p2GpztrdxpV8E8+K3jnR4wQHByrdfRh+TCsRvh5bLY3dpBeMsUsMkcAaFD5LMQwbOMtggYyfWtmw065/t+41W+SOOR7aO3RI3LAYyznOB/EQB/u578AGxRRRQAUUUUAFcRD431CKw0/XdWtLK38PajIPJnjld5ljkz5Dsm3HzDaSATt3e1dvWPa+F9Ns9R+1wrMdrF4rdpmaGFj1ZIzwp5PT1NADR4v0NtOa/F9/oyyiIyeU/DEZxjbnpWJ4h8Q6Wmq+E9Wa6xYtdXAEvltz+4kHTGeo9K7Wud17/AJG/wt/19XH/AKTSUAXLrxRo9lY215c3myC6BML+U53AewGR171jeNNWskvtF0tpsXj6raSLFsblfM65xjse9dfXO+Mv+PbSP+wxaf8AowUAWfFl3eWfh55dPkMMhmhR5lXcYo2lVXcD1Ckn2684rhbi+u28VrJdaxcpaWzXUVpfLCu+QeVE20nbtOGLjoM4x15r1MgEYIyDTDDGUCGNCq9F2jAoA8zj8U+Ik1BJHeW4kkiPl2cSKAH+zswDqRuGWUHcGI5xjmix1vxFdWMCQalM32mazV52RHeLzJCHwAoAGAeDkjFdPc+M7G28RXenm0LTWsbFzvUS4CF8iM/MUwMbhkZI+tVdV8ZLbT6Zb6RZs8t49tNs2qA0cpI65wG46/zoAyL7XfFVnHqFoju76UD5tyYh+9SSVVRumMrHvYgDrj6U7TtV1291KztP7TuDaNdy7pkRHkaNYA+wsFC/f7gd8dq05fiJp1rJdtfK0HkRokkDlFxKZWjK72YD+HPOBgE+1SWHjyz1AfZ9I06S8uYPMNxDaujpEEKhirjhs7wQByec4IIABxrjydPuY9O+zXbS2d2Iru0ga3vbc+WxxKB97JGOTnOCB3GjN4l1JdXtYrTU7x7ZZrW0fcI8SB4lLSfd3H7w+bIGc8cV1WgeKLe8jv59Vjt9KZL6W2jM0ioZlQgAnJznkA+9XX17Rre5jjuNsEUoUW900Y8iYHoFlGVHPABIPoDQB53a654gtbC1httSnKx2ytA85Q/aJjKVKPlcnsMDBGc16Hr+oanYtB/ZtsZw6tvxCz4PGPu9O9c9/b10bGDXZby1tLWS6KxWktsTGsQkCMzSKpZHwcgnAz271Nf/ABK0+E3kNmsctxC1xDEhnRmeWEsGDIG3KuUYAnHSgDrrCWaawhkuV2SsuXXaVwfoelWK8/vPF2qW98k2oPLpUDpA9t5lmXtZS6ruWWdQwjOSQMle3WrP/CwxZ2N3datZx28cF9c26tJcpECsTEZ+Y8sQBwOuaAO3orif+FixQXFyb21WC3aaGOzlmlWJZBJD5uXZjhcAH9BU9r4vbXxNLoUirFa2TXEoZQxL+Y6BcjIxmGUZHXAxQB19cz4hghPjXwncGJDOtzcIspUbgptpCQD1wSAcewrfsbpb7T7e7RdqzxLIFznGRnFYniD/AJGzwp/1+T/+kstAHRUUUUAFFFFABRRRQByHiO7vLP4g6DJp+nPqMp02/UxJKkZA8y0+bLED0/Orv9u+If8AoULn/wADrf8A+Kov/wDkpehf9grUP/RtpVzWdaksLuzsLG3W5vr0t5UbybFVVGWZjgnj2FAFP+3fEP8A0KFz/wCB1v8A/FUf274h/wChQuf/AAOt/wD4qpx4kSzt869bPY3BnMMcMQa4M+Bu3RhF3MNvJ+XjBz0rN1D4h6XZyXKRZdYYYZRdPlLciViAWkxhcY5oAt/274h/6FC5/wDA63/+Ko/t3xD/ANChc/8Agdb/APxVRJ4x8i40+DU7IqdRvPsltPZsbiGT9y0m/cq8D5CDnpyc4BIdc+PNFhsftUL3FwvmRptS1lDESHCuAV5Q/wB4ce9AD/7d8Q/9Chc/+B1v/wDFUf274h/6FC5/8Drf/wCKqaXxhoscVy4umYW8Etwf3LgOkf3yjEAPjjO0nGRVrT9UluL6WyvbcW9zHDHNtDbgwYcgf7rDB+ooAz/7d8Q/9Chc/wDgdb//ABVH9u+If+hQuf8AwOt//iq6FmCIWPQDNcrZeNXa0sNS1mxg07SNTEf2O7N5vJ8wAxiRSi7CwIHBYA8Z70AWf7d8Q/8AQoXP/gdb/wDxVH9u+If+hQuf/A63/wDiqvReJ9Am1k6RDrmmyamGZDZJdxmYMoJI2Z3ZABJ47UWvifQL7VX0yx1zTbm/jLK9pDdxvKpX7wKA5GO/HFAHJ+OtZ1yX4eeIo5/C1xBE+l3KvKbyBggMTZbAbJx1wK9ArjPGPiDRtZ+G3i6PSNXsL97bSboTra3KSmLMT/eCk46Hr6Gt228V+Hb2wub6z1/S7i0tADcXEV5G0cOem9gcL+NAGt061m/8JJof/QZ0/wD8Ck/xqpJ4z8Lto8moL4k0g2QfyDci+i8sSFchN27G7HOOuKzvAuh6TL8O/Dkkml2bu2lWrMzW6EkmJcknFAG5/wAJJof/AEGdP/8AApP8aP8AhJND/wCgzp//AIFJ/jTv7A0f/oE2P/gMn+FH9gaP/wBAmx/8Bk/woAb/AMJJof8A0GdP/wDApP8AGj/hJND/AOgzp/8A4FJ/jTv7A0f/AKBNj/4DJ/hR/YGj/wDQJsf/AAGT/CgBv/CSaH/0GdP/APApP8aP+Ek0P/oM6f8A+BSf407+wNH/AOgTY/8AgMn+FH9gaP8A9Amx/wDAZP8ACgBv/CSaH/0GdP8A/ApP8aP+Ek0P/oM6f/4FJ/jTv7A0f/oE2P8A4DJ/hR/YGj/9Amx/8Bk/woAb/wAJJof/AEGdP/8AApP8aP8AhJND/wCgzp//AIFJ/jTv7A0f/oE2P/gMn+FH9gaP/wBAmx/8Bk/woAb/AMJJof8A0GdP/wDApP8AGj/hJND/AOgzp/8A4FJ/jWT4o8B6P4h0KSyjsrW0nDLJBPFAoKOp9Rg4IypwQcMcEHmvMo9J+waJDqWq6ZavYsGE14kKBbSRGKPHN7q6lTIo2HBPyjGQD2e31vSbudYLTU7OeV/uxx3CMx4zwAavV4xpFpZReOPCs1pbwJuv5MPEgGQbS47jtXs9ABRRRQAUUUUAFFFFABXHa3eX1l8SLB9O0uTUXbSbgNGkyRlR50POXIFdjXOzf8lPs/8AsD3H/o6GgA/t3xD/ANChc/8Agdb/APxVH9u+If8AoULn/wADrf8A+KrT1LWrDSDGL6ZlaUMyJHE8jFVxubaoJCjIyegyPUVl2njfSbjUL61ml8hbW5W3ScqzQy7oUlVhIBsGQ+AM5OPegBf7d8Q/9Chc/wDgdb//ABVH9u+If+hQuf8AwOt//iqsR+LtFlgMqXUmNyKqG2lDyFs7diFdzg4JyoPAJ6A02TxnoEUcbyagoWRS3+qclAG2NvGMphgVO7GCMHBoAh/t3xD/ANChc/8Agdb/APxVH9u+If8AoULn/wADrf8A+Kq7eeJtJsJWinui0ykjyoYnlc4UMSFQEkAEZIGBkA8kVQ1bx1ounW8rQ3kNy8aRuWVj5SK+NrPKAVVSCDnPI6UAO/t3xD/0KFz/AOB1v/8AFUf274h/6FC5/wDA63/+KqKPxiYJ7CHUrEj+0LwWltPZObmGTMbPv3KvA+XBz05OcAkOuvHuiw6e13A9xcKrRjalrKGZZDhXUFfmQ/3hke9AD/7d8Q/9Chc/+B1v/wDFUf274h/6FC5/8Drf/wCKq/aeItLvr4Wdrcl5mDFcxOqvtIDbWICtjIzgnGa06AOd/t3xD/0KFz/4HW//AMVR/bviH/oULn/wOt//AIquiooA53+3fEP/AEKFz/4HW/8A8VWVdajqd7428MpqOhy6ciz3BWR7iOQMfs78YQk129c7r3/I3+Fv+vq4/wDSaSgDoq53xl/x7aR/2GLT/wBGCuirH8S6Rd6xp9vHp9zDb3NvdxXUbzxGRCY23YIDA8/WgDYornfI8af9BLQf/BfN/wDHqPI8af8AQS0H/wAF83/x6gB+oeD7DVNUW8vri7mVWZ1t2dTGpZGQkZXcPlY8A456VUHgCwxEz6jqTzwLEsNw0qb4xESUA+TBxnuDnvVjyPGn/QS0H/wXzf8Ax6jyPGn/AEEtB/8ABfN/8eoApL4EibUZ2muJPJMUXlTq4NwZllMhlYlduckcAYx2q9J4PgkMUh1XUxOgdXuBMvmSozBmRjt4XKjhduO2KTyPGn/QS0H/AMF83/x6jyPGn/QS0H/wXzf/AB6gC/p3h+y0w3XlBpVuZ2nKzbWCM3ULxnHfkmnSaDps1+Ly4tVuJVx5YmJdIsDA2IflU+4ANZ3keNP+gloP/gvm/wDj1HkeNP8AoJaD/wCC+b/49QBFeeA9PvYJrR77UI9PmmE5sYpVSIPuDZGF3Y3DON2M1Z/4RK2DXKi/1Bba4MzfZVmAjRpSS7D5c5JZjgkgE9OmI/I8af8AQS0H/wAF83/x6jyPGn/QS0H/AMF83/x6gBkngm0kgltjqOoC0uIliubYPHsuAF2nd8meRwdpANNuPAlhPdSzrfX8LSmfIjkTAWbBkUZQ4BIB9ffBxUvkeNP+gloP/gvm/wDj1HkeNP8AoJaD/wCC+b/49QBHD4FsLbDwXt8kymJo5g6b42jj8sEfJjlOCCCPYEU+bw1PDI7adc72nsmtZ5Lti7yHduDk+25+BgfNwAABS+R40/6CWg/+C+b/AOPUeR40/wCgloP/AIL5v/j1AG5ZWsdjYQWkP+rgjWNc+gGKw/EH/I2eFP8Ar8n/APSWWjyPGn/QS0H/AMF83/x6o49F8Q3ev6Zfa1qGmPDp8kkqx2lpJGzs0bR8lpGGPmz07UAdNRRRQAUUUUAFFFFAHO3/APyUvQv+wVqH/o20qzreizX93Z39hJDFeWm8Dzk3JIjDDI3tVa//AOSl6F/2CtQ/9G2lL4tvNTsoLR9OivHtzIRdPYxLLMi4+UhG6jPXGTQBhRfDd4Xmu1ubX7S98bxYYYTBCMwrGUwhz/CDu785HJq/beC7rTWY6TqEUGYIoyphJV2RmY5GfutvbPJP1qhH8Q4rGwiTzk1WZIXmlllzauUV2XbsK/635GBUheR2pJvE/iC7kvjBaS/YrbU2gL2AWSfyfJRwQjgZ5fnGTz7cgFhPAU0Tx3Vpc21lc/2h9tMdrEUiTNtJAdoz97EpbOBkgD3qvZfDq6t3aWW9haZo4Qz/ADu0jxsG3M7Esc8/So7bxvqME0DDytRsf7Ou7mSaf/RJd8UiqFZXUBSNwDE47nHGDYX4hTTGGaKyh+zRTzR3jJcFxtSLzNyHYN3HrigCMfDUx6be2a3ELRvaXENuSJCytKOCSWIAHsOfboem0+1up9el1W8gNvm1jhSIvkgn5nzjjg4H4GueXx7Jqdr9ns4I4Lq4mtYoJIZvOUJNIVJJ2jDqqudvPQc9cdJouoy3V3qllckNJZXWxWHVo2UMpP5kf8BoA1iARgjINYdj4Us7DUobqO4unitQws7N2XybTcMERgKGxjIALEAcAAAY3KKACiiigDnfiH/yTLxP/wBgi6/9EtXRVzvxD/5Jl4n/AOwRdf8Aolq6KgBk3+of/dP8qw/AX/JN/DX/AGCbX/0Stbk3+of/AHT/ACrD8Bf8k38Nf9gm1/8ARK0AUx8QdLaKKQQ3AE3mNGHULvSPAZhk9MsAO5NSaR42tdU1WS1WF1id4xazDlZleLzA3txSy+CLP7NZx2lw8ElojxJKY0kOx2DEYcEdQMHrSSeCYBffa7O/uLWRXiePaqMEKJs7jnK+tADbn4g6VbI8ghu5YordbmZ44SRGjOyDPqSyEYFadv4htpbTUJpY5YG07P2iN1yVwu7jHXg1nQ+BbCKwntHuJ5EngigdiQDhJGkB4HXLGtWHRIIJtUlilkV9ScPIwP3CECjH4CgCnpni+w1P7MI1kRrmUxRZAIb92ZM5HH3Qfx4rMh8fwyaxeRm0nfT4IoH+0JH/AKvfJIh3fig6epp9v4AgtGE9tqVxFeLcfaFnSKNQD5bRkbAu3kMecZzzUVt8ObaxjeO21O88iWJIp4GK4nCSO4y2MjmQ5IoAf/wsLTrW0El6/mFI2nmaIYEcQcruIJyfuscD+6ak1Xx3DZafdXNpp9zcpEzpHIFxHKyPscZ7YOR7kVT034eqthEbqYWly8Xk3C26rKrJvZgAzrkH52GRg8+1aUngqKWxu7A6ldCxnMjR24ChYWdy5IOMn5iSAcjmgCh/wmF2fEX2OWS3tHM8ccVhcxMr3CMRl1lztzgkhcckYzzUlt8Q7X+xxfalZTWo+03MTA44WKZot3OMk4HA9/SrbeDfMZY59WuprP7Qly9tIiEM6uHyGxuUFlHAIFUJ/hrZ3BmD6jcGOVpzsaONggllMp25XghmOD1oAnj8f2qy3MV1bPHKly8MCBh++VVDF8nAGAw/T1pujeHUu9LjvdD8U61bWN673ccMf2YqhldpGA3Qk/eduCTipn8B2plaaO8lWfzTIjvFHJs3IquMMpBB2g89O1dDp1kmnabb2cbs6wRhN7Yy2B1OO560Aebf8IAvhXx54bv7PWr26trnVHD2t2qEJIbS5O5CgUIuBjYFx34r1Kud8T/8hvwl/wBhhv8A0iua6KgAooooAKKKKACiiigArnZv+Sn2f/YHuP8A0dDXRVy+stqFh41stUtdFvdTtxp81u/2NoQUdpI2GRJInGFPTNAF/V9IvbnUYdQ0m7itrlIHt286MupViCDgEcgj8c1hWXgCawtzp0N/GdN+0Q3AV4iZAYoI4lGc4IPlAnPvWp/wlGo/9Cbr3/fVp/8AJFH/AAlGo/8AQm69/wB9Wn/yRQBiQ/DiSP5vtqRqkyTJaxPKIS4WRWbG/KFhL0XAG0cVrWPgyK3ikjmlQpPaSwSJGpA3SSFywyT3Pfnipf8AhKNR/wChN17/AL6tP/kij/hKNR/6E3Xv++rT/wCSKAMo/D5obWwa3u0lvbUTCSWbeiymQqdxCMDwUXj0yKvWvhGfTLOeLS9QWKR1tlQtDlNsKBNhXP3WA6Z7/nP/AMJRqP8A0Juvf99Wn/yRR/wlGo/9Cbr3/fVp/wDJFAGWngKaKRbq2uLWyuft4vGjtYikS/uHhOBn72JC2cDJAHvVWz+HN1AzSTX0LzNBEjyHe7yOjht7O5J5O76cYre/4SjUf+hN17/vq0/+SKP+Eo1H/oTde/76tP8A5IoAp+H/AAR/YmqpcGaKSGAymIYcvlyDyWYgY56DnPbv11c7/wAJRqP/AEJuvf8AfVp/8kUf8JRqP/Qm69/31af/ACRQB0VFc7/wlGo/9Cbr3/fVp/8AJFH/AAlGo/8AQm69/wB9Wn/yRQB0Vc7r3/I3+Fv+vq4/9JpKP+Eo1H/oTde/76tP/kiqb3Op614r0KZvDupafb2Us0k0121vtAaF0AASViTlh2oA66iiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOdv/wDkpehf9grUP/RtpWnqejW2qPBLK00U9sxaGeCQo6ZGD7EEdiDXP+I11N/iDoI0Wa0iuP7Nv9zXcTSJt8y0yMKynOcd/Wrvk+NP+f8A0H/wBm/+O0ATL4R0uNQIftEJIdZXjuGVpg7bm3nOTkknPUZOCM0S+EtOku5LqJri3uHn89ZIJNpRygQ47YKqAQcjiofJ8af8/wDoP/gDN/8AHaPJ8af8/wDoP/gDN/8AHaAFm8EaNPAkTxzDasiswlO5zI6yMzHrneitn1HpxTv+EM0pnkeX7RK0shlkaSUsXYpsbOexXgimeT40/wCf/Qf/AABm/wDjtHk+NP8An/0H/wAAZv8A47QA+XwpAmmPBaXE/nK0LwyXEhkEZhcPGv8Au5GD3IJ5rQ0vTTYPezSSeZNeXDTOR0HAVQB/uqPxzWZ5PjT/AJ/9B/8AAGb/AOO0eT40/wCf/Qf/AABm/wDjtAHRUVzvk+NP+f8A0H/wBm/+O0eT40/5/wDQf/AGb/47QB0VFc75PjT/AJ/9B/8AAGb/AOO0eT40/wCf/Qf/AABm/wDjtAB8Q/8AkmXif/sEXX/olq6KvP8Ax1D4uHw88RG7vdFaAaXc+asVnKrFfKbIBMpAOPUGvQKAGTf6h/8AdP8AKuT8C67pEXw78ORy6rZI6aVaqytcICCIlyCM119Z3/CO6J/0B7D/AMBU/wAKAD/hIdF/6C9h/wCBKf40f8JDov8A0F7D/wACU/xo/wCEd0T/AKA9h/4Cp/hR/wAI7on/AEB7D/wFT/CgA/4SHRf+gvYf+BKf40f8JDov/QXsP/AlP8aP+Ed0T/oD2H/gKn+FH/CO6J/0B7D/AMBU/wAKAD/hIdF/6C9h/wCBKf40f8JDov8A0F7D/wACU/xo/wCEd0T/AKA9h/4Cp/hR/wAI7on/AEB7D/wFT/CgA/4SHRf+gvYf+BKf40f8JDov/QXsP/AlP8aP+Ed0T/oD2H/gKn+FH/CO6J/0B7D/AMBU/wAKAD/hIdF/6C9h/wCBKf40f8JDov8A0F7D/wACU/xo/wCEd0T/AKA9h/4Cp/hR/wAI7on/AEB7D/wFT/CgA/4SHRf+gvYf+BKf40f8JDov/QXsP/AlP8aP+Ed0T/oD2H/gKn+FH/CO6J/0B7D/AMBU/wAKAMPXtV0698Q+E4rK/tbiQauzFIplc4+x3POAa62qVvoul2s6zWum2cMq/dkjgVWHbggVdoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDnb/wD5KXoX/YK1D/0baVP4i8Tw+HmtlltpJ2uCwUK6oBgZPLEDPt1qC/8A+Sl6F/2CtQ/9G2lXNd8PQa9Gsd1NKkYVkdEI2urDByD39D1FAGfqHjK3t9Jubi3ikMsc7WqBxx5gjL889OKp23jZ/wCz9LOqQmwurpIJW3qGWWNh8zLhvl59ckehq2fAem+eT5915HmiZbfzPlEnl+Xu/wC+SfzzVabwKgXT4ormS4jtpYlJuGyUgjB/drj14yT6UAFr8R9Ou1IitJy8gj+zIrITcb5BGoGD8vJGd2MA5qK+8Y6wL6SzstCmjlFhJcfvirFWSVUwFDfMMHPB5rQj8D6dFA8SSzqo8vyNrAGAxuHQrjvkDr1xilk8E2MzK013ePJ5TxPIZjucM4fOexDKMY9MUAZEHjO8jgzFPDqzNd2kBQQG2lgEsmw70Zjxxwc9+nGTJq3xEFpFqEVpp0jXdrEZY0eRMSKJAhJw3y9c4OK1J/BsF35kt3fXUt27QEXJI3KIZPMRQMYxuzn6mqo+HemCN0FxchWhaBQGA2oWDAdOSCByeaAJLDx3ZXutRaabeSN5J2tRIWXHnKrMVwDnHyN82MZx6iuprBtPCVlZ6ul/FJLlZHm8rjaZWBDOeM85PHTJreoAKKKKAOd+If8AyTLxP/2CLr/0S1dFXO/EP/kmXif/ALBF1/6JauioAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOe13TdYfxHpmr6HHYzNa21zbyRXkzxAiVoWBBVG6eSeMd6T7R40/6Bmg/+DGb/AOMV0VFAHO/aPGn/AEDNB/8ABjN/8Yo+0eNP+gZoP/gxm/8AjFdFRQBzv2jxp/0DNB/8GM3/AMYo+0eNP+gZoP8A4MZv/jFdFRQBzv2jxp/0DNB/8GM3/wAYo+0eNP8AoGaD/wCDGb/4xXRUUAc79o8af9AzQf8AwYzf/GKPtHjT/oGaD/4MZv8A4xXRUUAc79o8af8AQM0H/wAGM3/xij7R40/6Bmg/+DGb/wCMV0VFAHG6/Y+Mte8N6lpD2WhQLf2ktsZRfTMUDoV3Y8kZxnOK7KiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAP//Z)](http://laptrinh.vn/attachment.php?attachmentid=411&d=1414600939)  
  
Trong sơ đồ trên:

* Subject: các đối tượng liên quan đến nhau ( trong ví dụ ở đầu bài là các môi trường hiển thị dữ liệu).
* Observer: người quan sát, nhận các thông báo từ các đối tượng và gọi các đối tượng liên quan cập nhật.

**Áp dụng vào vấn đề đầu bài:**  
  
**Trường hợp 1:** có 1 Subject: dữ liệu giá vàng.  
  
Ta có các class sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | class GiaVang  {    }    public class WindowApp  {    }    public class Web  {    }    public class Mobile  {    } |

Ta xây dựng một abstract class cho các thiết bị, môi trường hiển thị:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | public abstract class Observer  {       protected Observer()       {         }         virtual public void Update()       {         }  } |

Nếu ta code cứng vào lớp GiaVang có 3 thuộc tính kiểu WindowApp, Web, Mobile thì sẽ khó khăn khi ta thêm một thiết bị mới để hiển thị. Nên ở đây ta sử dụng List Observer.Và Khi giá vàng có sự thay đổi thì lớp GiaVang sẽ gọi các thiết bị, các ứng dụng hiển thị cập nhật.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | class GiaVang  {         public static List Observers;         public void Attach(Observer concreteObserver)       {           Observers.Add(concreteObserver);       }         public void Detach(Observer concreteObserver)       {           Observers.Remove(concreteObserver);       }         public void DataChange()       {           for (int i = 0; i < Observers.Count; i++)               Observers[i].Update();       }  }    public abstract class Observer  {       protected Observer()       {           GiaVang.Observers.Add(this);       }         virtual public void Update()       {       }  } |

**Trường hợp 2:** có nhiều Subject: dữ liệu giá vàng, dữ liệu cổ phiếu…  
  
Nếu các thiết bị chỉ có thể hiển thị một số dữ liệu: ta tạo ra một lớp abstract và cho các lớp dữ liệu đó kế thừa. Và đẩy các hàm của lớp GiaVang lên lớp abstract này.   
  
[![Click image for larger version. 

Name: Observer_02.jpg 
Views: 23 
Size: 22.2 KB 
ID: 412](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDgRXhpZgAATU0AKgAAAAgABAE7AAIAAAAHAAAISodpAAQAAAABAAAIUpydAAEAAAAOAAAQyuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEh1bmdYdQAAAAWQAwACAAAAFAAAEKCQBAACAAAAFAAAELSSkQACAAAAAzk5AACSkgACAAAAAzk5AADqHAAHAAAIDAAACJQAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDEyOjA3OjA2IDE0OjM3OjQ0ADIwMTI6MDc6MDYgMTQ6Mzc6NDQAAABIAHUAbgBnAFgAdQAAAP/hCxlodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDEyLTA3LTA2VDE0OjM3OjQ0Ljk5MjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5IdW5nWHU8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgBygHJAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAorn77Up72aSGzmMFtGxQyx43yMOCFJ4Cg8Z5JIPTGTRNkjHLXN+T7ajcD9A9UoN6ickjrqK5IafF/z8ah/4M7n/AOOU4adF/wA/Gof+DO5/+OU/ZyFzI6uiuWGmw/8APxqH/gzuf/jlKNLg/wCfjUf/AAZ3P/xyjkkHMjqKK5n+yoP+fjUf/Bnc/wDxylGk2/8Az31H/wAGdz/8cpcjHdHS0Vzg0i3/AOe+o/8Agzuf/jlL/Y9t/wA99R/8Gdz/APHKOVhdHRUVz40W2/576j/4M7n/AOOU4aJa/wDPfUf/AAaXP/xylysLm9RWENDtf+e+o/8Ag0uf/jlKNCtP+e+pf+DS5/8AjlFmO5uUVijQbT/nvqX/AINLn/45SSaLp8MTSz3WoRxqMs76rcgAe58ylZhc26K5pY/DpI263Lu7f8TuY8/TzKuv9o0WI3K3Mt3YrzKkx3vEvd1bqQOpBycZweMEaa3EmnsbFFNjkSWNZImDowyGByDTqBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBxejMX0LT3P3ntYnb3JQEn8zXLarc61qHxObQbDXrnSrVdFF2v2eCByZfNKZPmRsSMY4BHSuo0T/kXtM/68of/Ra1mar4Oj1PxCdat9Z1PTLxrL7ExszDhotxb/lpGxByeoIPArptov66P9TO+/8AXVfoZfhL4hDU7TTbTV7dxqdyl3ueBB5LfZn2Owy2Ru4IHPfmjw740m8S+O7RbGSWPR7zQFv47eaNA4kM5XcSMnOBjG4ir8nw90hYtKTSprvSf7KilhgazZCWjkxvVvMVs5IyT1zzmoLD4bWmkzWE2j65q1hLZaeNOR4/s7l4g5f5g8TDOT1GOlCvdX/rR/8AAE9ml/Wq/wCCZkXxT0vQra9/trULnULg6ndxQxNHb2rRxxEZUFpFVlXIAJbe2fu9cbY+JmhtZ3l1DHdzQ2mkJrDOka/PC27AXLD5/lPBwPemRfDXTra9W/07U9TsdQW6uLkXcLxFv35BkjKvGyFMgEZUkEDmpNZ+HVjrkksk+rapBJc6cNNu3heLN1EMkF90Zw2STldvX04qPe5V3t+j/W34j05n2v8Ar/lc6yzuUvLKC6iDBJo1kUN1AIyM/nU4qCztks7KC1iLFIY1jUt1IAwM/lU4q5WvoKN7ajxThTRThUFDxThTRThSAcKcKaKcKTGOFUtRVZLvTEcBlN2SVPQ4ikYZ+hAP4VdFQ3doLuNAJGikicSRyLztbBHTvwSPxpdQexyWk6ve3XiuSGfVYbgTs0ctsUIttq/wwPj5nXPzZ67u22ur0ZQ+iQo43LhlwfTJGPyquNHlEUcS3MKpEcxqtog8s+q+h5PIrStbdLS2SCLO1BgZ6mqqST2JinfU57wTO7aJZIx3BrONzn12r/jXT1yngf8A5BNh/wBeEf8A6CtdXWSNAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4rRP8AkXtM/wCvKH/0WtZ0H2/WtXvHi1i40+2sLoQC2tY4SZcKrMZDIjHndwF28c5JPGjon/IvaZ/15Q/+i1qK58N6bd6qNRZbmG6O3e9reTQCXb93esbqHx0+YHjjpXUuhizmr3x1eaJob3s1qmpFLu+EqKZBKkMM7qCqxwuMBQBucoucZbnNbMviq5i1LU7ZdNRvsVu00MPnP9pusBTlIvL+aP5sblZjnjbnipr3wZoWowmG6tZfLbzt6R3UsYkErF5A+1huBYk4OQM8AVZfw1p0t7PdyG9M00ZjLDUJwI1OM+WN+IicDlNppO/5/wDALbTd13/Us6FqLato8N5IbPc5YMLK5M8YIYjAcqpJGOQVGDkdq0hVLTNMtdJsha2SuIwxYmWV5XZicks7ksxJ7kmropshDhThTRThUlDxThTRThUjHinCminCkA4U4U0U4UmMcKeKYKeKljHCnCminCkByvgf/kE2H/XhH/6CtdXXKeB/+QTYf9eEf/oK11dJDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDitE/5F7TP+vKH/wBFrWgKpaFDI3hzSyqMQbKHnH/TNa0BBL/zzb8q6k1ZGVtRBThSiGT+435U4Qyf3G/Ki6CwCnCgRP8A3G/KnCN/7p/KldBYBThQEb+6fypwVv7p/KpuMUU4UgB9D+VKAfQ/lSuMeKcKYPofypwP1/KkA8U4UwMPf8qUMP8AIqRkgp4qIOvr+lPEi+v6UmMkFOFRiVPX9KUTJ/e/SkBzHgf/AJBNh/14R/8AoK11dcp4H/5BNh/14R/+grXV0kMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOe8Of8irpP8A14wf+i1p9vc6jdWcV1Fa2YimjEi77tgQpGRkeWcH8aZ4c/5FXSf+vGD/ANFrXOP4IV7yG8l0Tw/PewCMLey26mcmMAI28xkggKMc8YHpWivbQh7nST63aWegJq96ZIrdo0chImlf58BQFQFmOWAwAadY6za6ikDW8V6gnDlPtFhPARtIB3B0G3rxuxu5xnBrK8RaZct4Hj062ebz0a1jElugZ12yx5dQQRwATyCOORiqGuW3iSxMcOj39/qFymk3xS4nSPDT7oTFuCKke4DeFyBxnnqaOr/rpcqKvY6u5vra0ntYbiTZJdymGAbSd7hGfHHT5UY8+lLJe28V9DZySYnnR3jTafmC43HPTjcPzrzbQ7LVTLpssV7qGorHqZkCXmm31t9nP2S4UktdySOVLMg4O0HoMk5isre6maCVJfE1verpF0uoXd3bXUwguWEWTEh6nIfCwfKQOO1N6f1/X9W7hpp5/wCb/r7z1WisDwXJJJ4cXzbW8tgsrqn2ya4keVQfv/6RiVQf7rDjtkYY79DJQUUUUDCiiigAooooAKKKKAClH3h9aSlH3h9aAMLwP/yCbD/rwj/9BWurrlPA/wDyCbD/AK8I/wD0Fa6uskUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHP8AhtWPhTSMKf8Ajxg7f9M1rT2N/dP5VX8L/wDIoaP/ANeEH/ota1aaloKxS2N/dP5UbG/un8qu1wXinxvHpAub6S7+z21g7RlFwTK4OMEdycEAfj7jelTnWlyxMatWNJXZ2Gxv7p/KjY390/lSaHqQ1nw9p2qCPyhe2sVx5ec7N6BsZ9s1Jqeo2+k6VdaheFhb2sTSyFVLHCjJwB1PtWUm4tpmy12GbG/un8qNjf3T+VZFv4h1iC+tE8Q6Jb6fa3z+Vbyw3/nukhBZUlUxqFJAI+VnG7jOMGpPD/jTR9e0qG5XULGG6+xR3l1Zi7R3tEZA3zjggAMOSBS5t/ILGnsb+6fyo2N/dP5VUfxf4aj0mLVZPEOlLp0snlR3jXsYhd+flD7sE8HjPY1YfxBo0WqR6ZJq9gl/KxSO0a5QSuwUNgJnJO1genQg96LhYfsb+6fyo2N/dP5VHqHiHRdJtjcarrFhZQiXyDJc3KRqJMZ2ZYgbsc460278S6FYSzR32tadbSQKXmSa6RDGo25LAngfOnJ/vL6ijmCxNsb+6fyo2N/dP5VPaXltf2cV3YXEVzbTKHimhcOjqehDDgj3FTUcwWKWxv7p/KjY390/lV2ijmCxS2N/dP5UBGyPlP5Vdoo5gscp4H/5BNh/14R/+grXV1yngf8A5BNh/wBeEf8A6CtdXUoYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZXhf/kUNH/68IP/AEWtatYfhqdl8J6QABxYwf8AotambxFpysVbUbIEHBBnXj9aai2JtI1q4nXvhZofiHxXZaxfhmhty8ktljKTyMQQWOenByvfj3B6xbosoZdpUjII6Gl+0v6LWtOdSk7wdmROEZq0lcsKqooVAFVRgADAAqlremLrWhXmmtM0H2mFoxKgyYyRwwB7g4P4VL9pf0Wj7S/otZcrZdzCisfEmqXtmviCPSre1spRMXs5pJXu3AIUlWRREMndgGQ9BnjJwNN8D6smg6fpeq6fo0kOlaZJaRLbXcsZvGkRQwZhGDCuQSSu8scN8pGD3n2l/RaPtL+i0cr18xp2OCvfBniW/SwvNQmivrq3W4ga1GuXNmEikZCo+0wRq0uNmMNGMhhkkrubVt/BT2ul6xDbJbW8s9xbXFjtlZ/KaCGFY1dyNxAeI+uVPqSK6j7S/otH2l/Radnay/rr+fXcV1e5xOr+BdQmTSp7bZeXEEE8d5F/a1zpolkmZXeQSwAk/MpGxlwQRyNvNlfBk1npOrQ2dhaSyXNzazWkX9oSweUIYYUXM4jZ8q0TEfKc8Z6kV1v2l/RaPtL+i0WaVl/Wt/zC+tyLRIdRt9EtYtauVur9Y8TyqBhm+oVQcdMhVzjO1c4F6q32l/RaPtL+i0NNgtCzRVb7S/otH2l/RaVmFyzRVb7S/otAuHJHC0WYXOd8D/8AIJsP+vCP/wBBWurrlPA//IJsP+vCP/0Fa6ukhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBz3hz/AJFXSf8Arxg/9FrXNQeItUtVs4oNc0qGziihR7WXTnaYAKodfN88AHIbB2ccZBxz0vhz/kVdJ/68YP8A0WtaVXa8bE9bnLa1calY/DeF9HuFtNQMNrFDLJGHCMzonKnqOTVLWPFt48Wgf2Yxt5Lm5t2vhsVjGrTLE0RDdCWZxnH/ACzbpXWalp0WqWX2W4Z1TzI5MoQDlHDjqD3UVmN4P0wy3UiGdGur+G/kxJnDxsGCgEHClgWIHd2PGapfFd7XQn8NlvqYd58VNI+2alp2lKby/tYpzEqXFu3myRAlk8vzRIuNrcuqg7TgnK5RfHV5aajNPqWk6iLRdKtryaCNIT9kDPOJJXYPgrhEO1WZsDheGrct/CqW0l3HHquoHTroS7tNYxGFDISXKt5fmDlmON+BngYwAxfB1qdOvbW61C9ujfacumzTymMP5S+ZgjagUNiVhnGOBx1yo+fl+t/0K02/rdfpcddeL7e21GWAaffzWltKkN1qEaJ5Fu7BSFYFw54dMlUZRu5Iw2KFz4wkn1bT4bGyv4bKTUXtnvniQw3ASOXeqjcXGGTqVUHbwTkZu3Hg+3uNRmnGpahFaXMqTXOno8fkTuoUBmJQuM7EyFdVO3kHLZbD4Lt4NQhnj1XUhbW1093bWO6PyYZHD7iPk3kZkY4ZiBnjAAFHb+u3/Bt+Iuj/AK/r+rF7w7ryeJNKXUbeyuLW2lw0DzvE3nIRkOvlu+B7Ng+orVrH0Hw7FoUl7N9suL65vpBJPcXCRIzkLgZESIucd9u48ZJAAGxTEFFFFAwooooAKKKKAClH3h9aSlH3h9aAMLwP/wAgmw/68I//AEFa6uuU8D/8gmw/68I//QVrq6yRQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAY/hiFG8I6OTnJsYP8A0Wtaf2dPf86z/C//ACKGj/8AXhB/6LWtWhNgReRGBzn86zr7VLC0tvMikS5ckqkccg+ZgMkZ7YBH5itOVPMjK5K57gkfyINcp4m8L3t9aQ3VldF72zZ3iTc2HVguV+Zm5+QY7ckYGc1dOzmlPY0pRg5pTeh0NjNa6hbCaAnHRlJ5U+hqx9nT3/OuK+Gej65a217rHiT91c6mItlqU2tCib8ZHbO88dRgZ5JA6vXNTGj6De6h5Zla3hZ0iHWR8fKg9ycAfWiraEnyu6QVIRU3GL0JYGtLoSG1njmEbmNzG4ba46qcdCO4qX7Onv8AnXl9ouoWllrGj3Wmanosuq6Q0yyy/Zna4u40ImZfLeUZdWj4YA/KcezLbVbXSLuDV28X3d9AfDpeARNZsbryy+/YREAxjzk44GPmyM1nzO9v62l/8j+Jl0v/AFuv8/wPU/s6e/51GPsrXL2yyoZ0RXeIONyqxIViOoBKtg+x9K8t0vxrqcmlSRar4ntobIaksMviG2uILlIYjBvA87yI4eZAE3GMgZK5LYIv6zcX0Go6zrGieIpGWw8NWt4txHHBJ9v2NdMu9tu3YwBz5YUnIKlQMG1dlJcy03/4KR6P9nT3/Oj7Onv+dSI25Fb1GaWldkqzVyL7Onv+dH2dPf8AOpaKLsZF9nT3/Oj7Onv+dS0UXYEX2dPf86XyE9/zqSii7A5TwP8A8gmw/wCvCP8A9BWurrlPA/8AyCbD/rwj/wDQVrq6SAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMrwv/AMiho/8A14Qf+i1rVrK8L/8AIoaP/wBeEH/ota1aEAU15EijZ5HVEUZLMcAD60krMsTNGoZgOASR/IH+VcP4q1HUra1tZZY5ZbHzpFuCynjhNnVE45bsRnHOcYqnHnmoGlKn7Saje1zu+oyKK4j4c+KH8RHUo4I3OnWZiW2mdCC5bcWGfQAIQOoDe4A6bxFdTWPhjVLu0fy54LOWSN8A7WVCQcHg8jvRVj7NtPoOdJxqezNGiuOhm1nw9faa93rV1rtpqKOjQ3cVvHLFIsTSgxtGka4IQqQ+edp3DBzTi+J8WrWNnc+HNLnv9+pLY3MUVxaSmPMe/IdJzGSRjGGOMEEA4qXo7en4uxindX/ra53tFcXqXxJsLPUdY0mO3I1TTrWSdIpLm3YS7QuMrHIzxg71OXVeMntS602ueGNDbU7jxBd6jK91aJLCLOEIm+5jVxCqpvwVZlAZnbkYbIyRa28x9bHZ0Vy1z48tbLw/d6pe6Zd2psbkW11a3NxawvAxCsCXeYRYIdCMOfvDvkCGT4kaYLVLq00/Ub22NhFqUs9ukZSC3kLjexLjO3y2yF3EjlQ2DgSv/XzD+v0/M6+iubvvGttZahNANL1K5treeK2nvoI4zDHNIUCoQXDn/WISwUqM8nIIFfTfiRoGq+L38O2cwe5DyRrILiBg7x53r5ayGVcYbl0VTt4JyuRK4X6nWUUUUAFFFFAHKeB/+QTYf9eEf/oK11dcp4H/AOQTYf8AXhH/AOgrXV0IAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAyvC/8AyKGj/wDXhB/6LWtWsrwv/wAiho//AF4Qf+i1rVoQBTJoY7iB4Z41kikUq6MMhgeoIpZHEaFiCcdlUk/kK5TxD4nksktraRWtTdtIvmrvG0KF5yyrjJcc4PQ8+lQi5z5I7mlOnKpJRjuzotM0uy0bTYbDS7dLa1hGEjTt+J5JPUk8mnajYx6npd1YTs6xXULwuyEBgGBBIznnmsfwv4mh1ya9sQ4luNPEfnSLja2/djp/ENhz/kDV1e//ALL0W91Dy/N+yW8k3l7tu/apOM4OM464pVIuLamEoThU5XuYLeA7a7sriDWdY1TVWltWtYprpoVa2Rhhtixxom44HzMrHjHQkF6eCY1hmMmuarLey3UV0b9/I81XjXauFEQjA28EbO5PXmn6d4i1YalaWniPR7awW/Qm1uLO+NzGWClij7o4yjbQSMBgcNyDgEufiD4UtrW1uv8AhINNmtrm7+yLPDexMiybdxBbdjgYz3+Ycc0Na6+X330/EyVrabf8D/Iht/h/p8VxIbq+v76zZblEsLho/JjW4bdKBtRWbJPVmYjseTmdPB4a3+z32vavfwJLDJBHcyRYh8qVZFAKxgvyiglyzYzyCSTp2uv6Pfajc6fZatY3F7aDNxbQ3KPJDzj51Byv41mXPjrQkhsbqy1TTr2wuLx7We9hvUMVttgkmJZgSOBGMgkY3Z+qXdFa/d/w4uoeDbe8vpL621K/0+8e9+2ie28pij+QICAJI2XBRR1BOc4I6UyDwLpkGk3mnrcXjRXmmLpkjPIpcRL5uGB2/f8A3zcnPQcdc79peW1/ZxXdhcRXNtMoeKaFw6Op6EMOCPcVNT2BSe6PPtb8Oahc+JGtdKttZgsbm+trq6kM9qti5jMbM+ATcbiIwu0AIW5Ixknp9L8OHSNRlmtNX1D7FI8kg01/JaBGdizFW8vzPvEnG8gZwBjAG1RRfSxNgooopDCiiigDlPA//IJsP+vCP/0Fa6uuU8D/APIJsP8Arwj/APQVrq6EAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZXhf8A5FDR/wDrwg/9FrWrWX4b/d6BbWjcS2SC1kXupQbf1ABHsQa1KEAjKGUqwBB6g1m6l4e07U9NkspbZI0Zt6tEoVkfGNw9/wCfQ8Vp0ULR3Q02ndHPeDfBth4M0f7JZkzTyYNxdMuGlI6cdlGTgdsnqSSdPXLGXU/D2o2NuyLLdWskKFyQoZlIGcZ45q9RRJud79R88ubme5w+o+F9f8VaU9nr50/TEhtZIrZLG5kuC0rxmPzHcpEQArMNqjndncMAVDY+EdWtLVbqPT7SLU47yCYrN4gu71Z0QMMGWaMtHjzGIAUg9yM5HfUUdb+n4O5Fla39djza58B+ItW1O+l1O9gQT215bC5GoXE4kWUjy/8ARmVYogqqqsEJLdS2eT0T6Tq+snRJdcsNLtX0zUftLQ2109wjIIJEUqWiTDB3BxjgLnOeK6eihaD/AOD+JleHNLm0fTJbadoyWvbqdfLJwEkneRRyByFYA++etatFFAdbhRRRQAUUUUAFFFV768jsLCa6mOEiUtjuT2A9yeB9aAOc8D/8gmw/68I//QVrq65nwlbtaWttbSffhtFjb6gKK6agAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAoXWnO9wbqwuDa3RADNt3pIB0DpkZ+oIPvjis+afxNC2NunuOxSCQ/+z1v0UAc39t8Tf88rP/wGk/8AiqPtvib/AJ5Wf/gNJ/8AFV0lFAHN/bfE3/PKz/8AAaT/AOKo+2+Jv+eVn/4DSf8AxVdJRQBzf23xN/zys/8AwGk/+Ko+2+Jv+eVn/wCA0n/xVdJRQByl7rHiGwspLmaKz2xjoLaTLE8AD5+pJAH1ptjrHia6SVZILFJ4JDFMiwSMFYc9d3Qggj2IrS1oXkmq2GzTbi8soCZ38h4hulHCAh3XgZLfUL6VW0u7udQ1u31Sz0u6gsb+1HnSTvCAwA3ROArls4JBGO49KQB9t8Tf88rP/wABpP8A4qj7b4m/55Wf/gNJ/wDFV0lFMDm/tvib/nlZ/wDgNJ/8VR9t8Tf88rP/AMBpP/iq6SigDm/tvib/AJ5Wf/gNJ/8AFUfbfE3/ADys/wDwGk/+KrpKKAOb+2+Jv+eVn/4DSf8AxVQva6rqEyPqW6TY2UiiiMcSn1IJJJ9ycegFdVRQBS06x+yRlpMGR+uOw9Ku0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVleF/+RQ0f/rwg/8ARa1q1leF/wDkUNH/AOvCD/0WtAGrRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVleF/8AkUNH/wCvCD/0WtatZXhf/kUNH/68IP8A0WtAGrRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVleF/+RQ0f/rwg/8ARa1q1leF/wDkUNH/AOvCD/0WtAGrRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVleF/8AkUNH/wCvCD/0WtatZXhf/kUNH/68IP8A0WtAGrRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVjW3iCe8tIbq10HUpIJkWSNw9uNykZBwZcjg96ANmisr+173/oXdS/7+W3/AMeo/te9/wChd1L/AL+W3/x6gDVorK/te9/6F3Uv+/lt/wDHqP7Xvf8AoXdS/wC/lt/8eoA1aKyv7Xvf+hd1L/v5bf8Ax6j+173/AKF3Uv8Av5bf/HqANWisr+173/oXdS/7+W3/AMeo/te9/wChd1L/AL+W3/x6gDVorK/te9/6F3Uv+/lt/wDHqP7Xvf8AoXdS/wC/lt/8eoA1ayvC/wDyKGj/APXhB/6LWj+173/oXdS/7+W3/wAeqno13qOnaDYWU/h/UGktraOFyktsQSqgHGZenFIDoaKyv7Xvf+hd1L/v5bf/AB6j+173/oXdS/7+W3/x6mBq0Vlf2ve/9C7qX/fy2/8Aj1H9r3v/AELupf8Afy2/+PUAatFZX9r3v/Qu6l/38tv/AI9R/a97/wBC7qX/AH8tv/j1AGrRWV/a97/0Lupf9/Lb/wCPUf2ve/8AQu6l/wB/Lb/49QBq0Vlf2ve/9C7qX/fy2/8Aj1H9r3v/AELupf8Afy2/+PUAatFZX9r3v/Qu6l/38tv/AI9R/a97/wBC7qX/AH8tv/j1AGrRWV/a97/0Lupf9/Lb/wCPUf2ve/8AQu6l/wB/Lb/49QBq0VBY3keoadbXsAYRXMSyoGGDtYAjPvzU9ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFZXhf8A5FDR/wDrwg/9FrWrWV4X/wCRQ0f/AK8IP/Ra0AatFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGV4X/wCRQ0f/AK8IP/Ra1q1leF/+RQ0f/rwg/wDRa1q0IAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACsrwv/yKGj/9eEH/AKLWtWsrwv8A8iho/wD14Qf+i1oA1aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMrwv8A8iho/wD14Qf+i1rVrK8L/wDIoaP/ANeEH/ota1aEAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAc/falPezSQ2cxgto2KGWPG+RhwQpPAUHjPJJB6YyaJskY5a5vyfbUbgfoHqHRmL6Fp7n7z2sTt7koCT+ZrmdY12+uvGF3o1tfXWmWWmWUd3cz2NoLm5maRmVURCkg2jaScISeOgBzsoxSV+v8Alczu3c60afF/z8ah/wCDO5/+OU4adF/z8ah/4M7n/wCOVxdn4rFxqHhkabrd7qtve215IFWxiQ3/AJQGNzPs8tgeBhQGPXaKoXfxNXUDaahoUslvaabqsNprEEpglVophtVxJGzrhWI+63Yg1XLG9rf1e35hd2v/AFtc9FGmw/8APxqH/gzuf/jlKNLg/wCfjUf/AAZ3P/xyuL0zxneQ2PiHxDqYlutIh1NrSziRreAQxR/I0heV0BDSAjliemBSn4waI1gLyz0zVbyEaaNTlaCOL9zDvZDu3SDkMpyBn2zU2ja/9bX/ACHr/Xrb8ztf7Kg/5+NR/wDBnc//ABylGk2//PfUf/Bnc/8AxyuU1H4p6RYfaZYNO1PULK0eCO4vbRIjFE8wUop3SKxOGUnCkDIruBRyrsJSKg0i3/576j/4M7n/AOOUkOhWUEKQwyX8ccahURNSuAFA4AAEnAq+KcKXKh3ZSGi23/PfUf8AwZ3P/wAcpw0S1/576j/4NLn/AOOVeFOFKyHcoDQ7X/nvqP8A4NLn/wCOUo0K0/576l/4NLn/AOOVoCnClZDM8aDaf899S/8ABpc//HKSTRdPhiaWe61CONRlnfVbkAD3PmVpiqWoqsl1piONyG6JKnkHEUjDj2IB+oFKyE3Yz1j8Okjbrcu7t/xO5jz9PMq6/wBo0WI3K3Mt3YrzKkx3vEvd1bqQOpBycZweMHndL1q8vdZgsLnVxPa+dIqqLbFxPjqkybcRhe7YGcr05rqtGUPokKONy4ZcH0yRj8qqdPlFGVy9HIksayRMHRhkMDkGnVzHgmd20SyRjuDWcbnPrtX/ABrp6zLCiiigAooooAKKKKACufvtSnvZpIbOYwW0bFDLHjfIw4IUngKDxnkkg9MZPQVxejMX0LT3P3ntYnb3JQEn8zVwipPUmTsiY2SMctc35PtqNwP0D0o0+L/n41D/AMGdz/8AHK5i81HU9Z8ez+HtP1GXSbaws47qee3jjeaZpCwVB5isqqApJ+XJOOnOZtS8Z2/ht20+5g1HWLmwsRd39xbRRDyohx5jgsgydrHagJ4PHStbQtf+tL/5E3e39f1qdGNOi/5+NQ/8Gdz/APHKcNNh/wCfjUP/AAZ3P/xyvKtR8YeI7/xTrcXhbUdQuTbtYPpdlBpyyW0ySorv50pizGpBJyXUjnHTFbHhr4hzi6ubLxDFdubjXLrT7O+WOEQoUyUiIDBycKedpByMmptHt/Wn+YO61v8A1a/6HfDS4P8An41H/wAGdz/8cp39lQf8/Go/+DO5/wDjlecaf8Y9D0jwzpja3qN1qV3NbfaJp2it7aRUMhUMYvMAJ4Pyx72wuSOeeouviLo9pFqkjQ3jppl5b2cpRF+d59mwrluV/eDOcHrwafLG9l/WtvzC7Wr/AK6nQDSbf/nvqP8A4M7n/wCOU4aRb/8APfUf/Bnc/wDxyrYp4qeVdh3ZQh0KyghSGGS/jjjUKiJqVwAoHAAAk4FSDRbb/nvqP/gzuf8A45V0U8UuVDuyiNEtf+e+o/8Ag0uf/jlKNDtf+e+o/wDg0uf/AI5V8U4UrILmeNCtP+e+pf8Ag0uf/jlOGg2n/PfUv/Bpc/8AxytAU4UrIZmSaLp8MTSz3WoRxqMs76rcgAe58yqqx+HSRt1uXd2/4ncx5+nmVoaiqyXemI4DKbskqehxFIwz9CAfwrmdJ1e8uvFckM+qw3AnZo5bYoRbbV/hgfHzOufmz13dttXGmmmyHJ3Ojf7RosRuVuZbuxXmVJjveJe7q3UgdSDk4zg8YOtHIksayRMHRhkMDkGqOjKH0SFHG5cMuD6ZIx+VZHgmd20SyRjuDWcbnPrtX/GsmrOxad1c6eiiigYUUUUAFFFFABRRRQAUUUUAFFFFAHFaJ/yL2mf9eUP/AKLWqep+FrfUNYj1e0vrzStSWHyGurLy90sWc7GWRHUgHkHGR2PWrmif8i9pn/XlD/6LWs6D7frWr3jxaxcafbWF0IBbWscJMuFVmMhkRjzu4C7eOcknjpSTSX9dv1MnoZc3wo0G40TT9Lknv/IsLS4tYyJVDMs332Y7eueRjA9QRxVmD4baWmm61Zz31/cjWoYobiWQxK0YjTahjCRqqkDHbsKo3vjq80TQ3vZrVNSKXd8JUUyCVIYZ3UFVjhcYCgDc5Rc4y3Oa2ZfFVzFqWp2y6ajfYrdpoYfOf7TdYCnKReX80fzY3KzHPG3PFDtr/XmNpp28/wCv0I3+HelnRdD063u722/sOTzbW4Ro2kLkEFn3oykncTnbnPIxVSy+E2hWWmXNjFd6i0VzpbaW7NIhYRNIzlh8n3sueemMcV0+hai2raPDeSGz3OWDCyuTPGCGIwHKqSRjkFRg5HatIUNb3/rS35CT/r0d/wAzynX/AIdXMtw+keHrfWYbO7ntGu7ia5tRZsIQn7zbzMX2oFwAqlsE8CvWhTRThS6DSXQeKcKaKcKkY8U4U0U4UgHCnCminCkxjhUV3aC7jTEjRSROJI5F6q2CPx4JH41KKeKQGWmiypeTXcd3GlxOoSWZLZQzAdOa07W3S0tkgiztQYGepp4pwpNt7gkkcr4H/wCQTYf9eEf/AKCtdXXKeB/+QTYf9eEf/oK11dSigooooAKKKKACiiigAritE/5F7TP+vKH/ANFrXa1xWif8i9pn/XlD/wCi1rWluRPYpap4Vt9Q1mLWLS+vNK1JIvIa6sjHuliznY6yI6sAeRxkdjVPUvAFnqjtI+rapBNcWIsL2WF4t17EO0mYzg8t8ybT8x9sWIPt+tavePFrFxp9tYXQgFtaxwky4VWYyGRGPO7gLt45ySeMa98dXmiaG97NapqRS7vhKimQSpDDO6gqscLjAUAbnKLnGW5zWtk0v63/AM7/AOYkn09P69Lf5HSaL4W0/QdU1C+07zUN+sCPEWBSNYU2IEGMjjrkms8fDnSD5H+kXv7jV5NYX505mcHKn5fucnjr71LL4quYtS1O2XTUb7FbtNDD5z/abrAU5SLy/mj+bG5WY54254rY0LUW1bR4byQ2e5ywYWVyZ4wQxGA5VSSMcgqMHI7UtJa/1v8A5onpb+tv8mc7pvwy0/RRbHQ9Y1bTpYbX7HJLDJEzXEe8uA4eNhkFmwyhTgnmptV+G2m6tqN7cvqWpW8d/PBc3NrA8XlySwldjfNGWB+UAgMAfSuuFOFJ9/67/nqMcKeKYKeKRQ4U8UwU8UgHCnCminCpGOFOFNFOFIZDeWgu40AkaKSJxJHIvO1sEdO/BI/Gqo0eURRxLcwqkRzGq2iDyz6r6HnqK0xThRdoVkMtbdLS2SCLO1BgZ6mua8D/APIJsP8Arwj/APQVrqhXK+B/+QTYf9eEf/oK1HUpHV0UUUAFFFFABRRRQAUUUUAFFFFABRRRQBxWif8AIvaZ/wBeUP8A6LWornw3pt3qo1FluYbo7d72t5NAJdv3d6xuofHT5geOOlS6J/yL2mf9eUP/AKLWsPxh4nvtH066WxsLqF42iVdRljjNsm51BZsuDgAkHj/GupW5UzLqaN74M0LUYTDdWsvlt529I7qWMSCVi8gfaw3AsScHIGeAKsv4a06W9nu5DemaaMxlhqE4EanGfLG/EROBym01R1DW7o/D/U9Wt1NtdQWU8kTgZXcqEh1DDlSRkZH1FUrnUtT8KTs13qU+uW8lhPdBLpYo5Y2i2nhoo1Gwh8HKkg455xS0G03Z9/8AgHU6ZplrpNkLWyVxGGLEyyvK7MTklnclmJPck1dFczfa/cDxRp+m248pPtqRzNw3mo1tPJjkcYaNeQe1VNV8aS6Rc6hBDZXF9crfrbQQhZHUD7PHKx/cwu6r8x/hc5PUA/Kf1+gKPb1OzFOFctF4xZ9OubmTS5oHt7OC5aGctG+ZGdShDKCMbOpGTnoK6kUmCHinCminCpGPFOFNFOFIBwpwpopwpMY4U8UwU8VLGOFOFNFOFIDlfA//ACCbD/rwj/8AQVrq65TwP/yCbD/rwj/9BWurpIYUUUUAFFFFABRRRQAVxWif8i9pn/XlD/6LWu1ritE/5F7TP+vKH/0Wta0tyJ7EVz4b0271Uaiy3MN0du97W8mgEu37u9Y3UPjp8wPHHSob3wZoWowmG6tZfLbzt6R3UsYkErF5A+1huBYk4OQM8AVneMPE99o+nXS2NhdQvG0SrqMscZtk3OoLNlwcAEg8f41c1DW7o/D/AFPVrdTbXUFlPJE4GV3KhIdQw5UkZGR9RWui2CKbaS6l5/DWnS3s93Ib0zTRmMsNQnAjU4z5Y34iJwOU2mr2maZa6TZC1slcRhixMsryuzE5JZ3JZiT3JNctc6lqfhSdmu9Sn1y3ksJ7oJdLFHLG0W08NFGo2EPg5UkHHPOK0L7X7geKNP023HlJ9tSOZuG81Gtp5McjjDRryD2pELVX+f3I6YU4Vxmq+NJdIudQghsri+uVv1toIQsjqB9njlY/uYXdV+Y/wucnqAfluReMWfTrm5k0uaB7ezguWhnLRvmRnUoQygjGzqRk56Cl/X6lNW3OpFPFMFPFIY4U8UwU8UgHCnCminCpGOFOFNFOFIY8U4U0U4VIDhXK+B/+QTYf9eEf/oK11QrlfA//ACCbD/rwj/8AQVpDOrooooAKKKKACiiigAooooAKKKKACiiigDitE/5F7TP+vKH/ANFrVq5tLe+tmt7yFJ4XxujkGQcHI4+oFQaZGbfTbe1cYe1QW7g9QyAL+uAR7EHvV0V1R+FGL3Iryyt9Q06exu499tcRNDKgJXcjDBGRgjg9qoWvhTSLa1urcwTXKXcRhma8upbl2jIwU3yMzBeTwCBk561rCnChpDuzBHgbQvs4iEV6CJVmEw1K5E29VZAfN8zfwrMvXocdhVmTwlo81q0Lw3GWlWfzlvZlmEixiIMJQ+8HYoUkNyM5zk52BThSBMxbvwbol/NDLdwXEskUKQbjezfvUQ5USfP+9IJJBfcQTnOa3hTRThSGPFOFNFOFSMeKcKaKcKQDhThTRThSYxwp4pgp4qWMcKcKaKivLuKwsZrqc4SJCx9T7D3PQUmBzngf/kE2H/XhH/6CtdXXM+Ebd7S1treT78NokbfUBQa6akMKKKKACiiigAooooAK4rRP+Re0z/ryh/8ARa12tcVon/IvaZ/15Q/+i1rWluRPYtXNpb31s1veQpPC+N0cgyDg5HH1Apbyyt9Q06exu499tcRNDKgJXcjDBGRgjg9qlFOFbMlO2qMm18KaRbWt1bmCa5S7iMMzXl1Lcu0ZGCm+RmYLyeAQMnPWoR4G0L7OIhFegiVZhMNSuRNvVWQHzfM38KzL16HHYVvCnipsgTMeTwlo81q0Lw3GWlWfzlvZlmEixiIMJQ+8HYoUkNyM5zk5Lvwbol/NDLdwXEskUKQbjezfvUQ5USfP+9IJJBfcQTnOa2hThQxjhTxTBTxUjHCnimCnikA4U4U0U4VIxwpwpopwpDHinCminCpAcK5XwP8A8gmw/wCvCP8A9BWuqFcr4H/5BNh/14R/+grSGdXRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGZqGj/AGqY3NpN9nuSAGLLvSTHTcuR+YIPuRxWNNaa7BJt8i1k/wBpN5B/TiusopptbCsjkPL13/n0g/J/8KNmu/8APpB+T/4V19FPmfcLI5HGvf8APpB+Un+FL/xPv+fS3/KT/Cutopcz7hZHJ51//n0t/wApP8KXdr//AD52/wCUn+FdXRRdhZHK+Zr4/wCXO3/75k/wpfN8Qf8APnbf98yf4V1NFF2Fjl/P8Qf8+dt/3zJ/hR9o8Q/8+dt/3zJ/hXUUUXYzmPtPiH/nztv++ZP8KX7V4i/58rX/AL5k/wAK6ailqBzX2zxEP+XK1/75k/wpftviL/nytf8AvmSukooA5z7f4j/58rX/AL5kqGS21XUJUOpZcIwKQxRFIwexOSST7k4HXAPNdTRQBS06x+yRlpMGR+v+yPSrtFFABRRRQAUUUUAFFFFABXM+HtKabwvpMgnC7rGA42Zx+7X3rpqyvC//ACKGj/8AXhB/6LWmm09AtcX+x3/5+V/79/8A16X+yH/5+F/79/8A1606KfNLuKyMz+yX/wCfhf8Av3/9el/suT/n4X/v3/8AXrSoo5mFkZ39mSf8/Cf9+z/8VS/2bL/z8J/36P8A8VWhRRzMLIof2fL/AM/Cf9+j/wDFUv2CYf8ALdP+/R/+Kq9RS5mFil9hm/57x/8Afo//ABVL9jn/AOe8f/fo/wDxVXKKLsLFP7JP/wA94/8Av0f/AIql+yz/APPeP/v0f/iqt0UXY7FX7Ncf894v+/R/+KpfIuB/y2i/79H/AOKqzRRcCv5Nz/z2i/79H/4ql8u5/wCe0X/fo/8AxVT0UgIQlyDzNFj/AK5H/wCKrmvA/wDyCbD/AK8I/wD0Fa6uuU8D/wDIJsP+vCP/ANBWgDq6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACsrwv/yKGj/9eMH/AKLWtWsnR5F09V0a4YJJBkW+eBNF/CV9SBwR14z0IoA1qKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArlPA//IJsP+vCP/0Fa2Na1RbC1MULBr2ZSIIwec9Nx9FHUn8OpFUvDVsLUJBGCY4IBGD9MAfyoA6CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqG6s7e+gMN3Ck0ZOdrjOD6j0PvU1FAGBP4Ut2Ytb3F0v+y11Kf13VD/AMIn/wBPM3/gXN/8VXS0UWA5r/hE/wDp5m/8C5v/AIqj/hE/+nmb/wAC5v8A4quloosBzX/CJ/8ATzN/4Fzf/FUf8In/ANPM3/gXN/8AFV0tFFgOa/4RP/p5m/8AAub/AOKo/wCET/6eZv8AwLm/+KrpaKLAc1/wif8A08zf+Bc3/wAVR/wif/TzN/4Fzf8AxVdLRRYDmv8AhE/+nmb/AMC5v/iqP+ET/wCnmb/wLm/+KrpaKLAc1/wif/TzN/4Fzf8AxVH/AAif/TzN/wCBc3/xVdLRRYDmv+ET/wCnmb/wLm/+Ko/4RP8A6eZv/Aub/wCKrpaKLAYNr4ZS2LbGRC/LPyzMfcnk/nWzb28dtF5cQIHU5PU+tS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB/9k=)](http://laptrinh.vn/attachment.php?attachmentid=412&d=1414601039)  
  
Ở đây, ta giả sử Mobile chỉ hiển thị được giá vàng, và 2 thiết bị còn lại hiển thị được cả 2 loại dữ liệu.  
  
Ta có các lớp Observer được cài đặt như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public abstract class Observer  {       protected Observer()       {       }         virtual public void Update()       {       }  } |

Và các lớp liên quan:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | public class Mobile: Observer  {       public Mobile()       {           GiaVang.Observers.Add(this);       }  }    public class Web: Observer  {       public Web()       {           GiaVang.Observers.Add(this);           CoPhieu.Observers.Add(this);       }  }    public class WindowApp: Observer  {       public WindowApp()       {           GiaVang.Observers.Add(this);           CoPhieu.Observers.Add(this);       }  } |

[Click image for larger version. 

Name: Observer_03.jpg 
Views: 22 
Size: 39.6 KB 
ID: 413](http://laptrinh.vn/attachment.php?attachmentid=413&d=1414601087)  
  
Như vậy,ở ví dụ trên ta đã giải quyết được các trường hợp:

* Dễ dàng thêm một thiết bị mới vào hệ thống.
* Dễ dàng cho phép các thiết bị hiển thị các thông tin khác nhau.
* Hệ thống được cập nhật thường xuyên,đảm bảo tính chính xác, nhất quán dữ liệu giữa các thiết bị hiển thị và cơ sở dữ liệu.