PART 6 (maybe last of this series)
JAVA: Visualize your OOP - by CodyScott
Para sa mga taga PHCORNER javadroids!

Hello, this is the part 6 and maybe last tutorial for this series.
Let's me talk a bit about our MainApp and also adding another CLASS to our system.

- let's add Vegetable class

Again, nothing new here. Pare lang sila ng Fruit class. Meron attributes na interesado tayo (name, type, etc.)...and also meron methods (setters and getters). Hopefully by this time, familiar na kayo. I would "Java semi-pro" na kayo... :)

MainApp
variables
methods

Basket	Fruit	Vegetable
variables	variables	variables
methods	methods	methods

Also, MainApp is where we "stage" everything or where we execute our "system".
- We create a Basket OBJECT
- We create Fruit OBJECTS and add to the basket
- New this time, we create Vegetable OBJECTS and add to the basket
- We perform query on the basket (number of items in, etc.)
- So ang basket natin ngayon ay hindi na siya SPECIFIC to Fruit only. Puwede na rin lagyan ng Vegetable. In other words GENERIC basket na siya.

TAKE NOTE OF THIS. Very very important.
Sa Basket Class, binago ko yung ArrayList to make it GENERIC... para ang basket natin ay puwedeng tumanggap ng iba ibang objects. Hindi lang siya pang Fruit, pang Vegetable din siya and other things.

```java
import java.util.ArrayList;

/**
 *
 * @author Cody Scott "CodyScott"
 * @version 1
 */
public class Basket {
    ArrayList<Fruit> fruitBasket;

    public Basket(){
        fruitBasket = new ArrayList<Fruit>();
    }

    public void addFruitToBasket(Fruit fruit){
        fruitBasket.add(fruit);
    }

    public void displayTotalFruits(){
        System.out.println("Basket has " + fruitBasket.size() + " fruits");
    }

    public void displayNameOfFruits(){
        for(Fruit thisFruit: fruitBasket){
            System.out.println(thisFruit.getFruitName());
        }
    }
}
```

this is exclusive for FRUIT.
Hindi siya flexible

```java
import java.util.ArrayList;

/**
 *
 * @author Cody Scott "CodyScott"
 * @version 1
 */
public class Basket {
    ArrayList<Object> basket;

    public Basket(){
        basket = new ArrayList<Object>();
    }

    public void addToBasket(Object object){
        basket.add(object);
    }

    public void displayTotalItems(){
        System.out.println("Basket has " + basket.size() + " items");
    }

    public ArrayList<Object> getBasket(){
        return basket;
    }


}
```

this is more flexible kasi we indicated "any Object". Object in java is the top of the hierarchy.

And this are just the methods or functions that we want the basket to do...for now.

You can add more if you want.

this is more flexible kasi we indicated "any Object". Object in java is the top of the hierarchy.

```
1  import java.util.ArrayList;
2
3  /**
4   *
5   * @author Cody Scott "CodyScott"
6   * @version 1
7   */
8  public class Basket {
9      ArrayList<Object> basket;
10
11     public Basket(){
12         basket = new ArrayList<Object>();
13     }
14
15     public void addToBasket(Object object){
16         basket.add(object);
17     }
18
19     public void displayTotalItems(){
20         System.out.println("Basket has " + basket.size() + " items");
21     }
22
23     public ArrayList<Object> getBasket(){
24         return basket;
25     }
26
27
28 }
```

Now, let's talk about our MainApp class.
- Personally, I like simplicity sa aking MainApp.
- Easy to follow, easy to read..... hindi magulo sa utak.
- Put all business logic to their own method

this is your application entry point.
dito lahat ang main processing from start to finish.
simple and it only takes 6 lines of codes (for this purpose)

```
4      public static void main(String[] args) {
5
6          initialize();                    //MainApp helper method (gawa gawa lang natin)
7
8          createFruits();                  //MainApp helper method (gawa gawa lang natin)
9
10         createVeggies();                 //MainApp helper method (gawa gawa lang natin)
11
12         basket.displayTotalItems();      //ready method from Basket class
13
14         displayNamesOfItems();           //MainApp helper method (gawa gawa lang natin)
15         displayCountOfEachItem();        //MainApp helper method (gawa gawa lang natin)
16     }
```

TALKING ABOUT BUSINESS LOGIC

Logic, validation, checking, etc.

These are your if then else, for loop, while loop, error checking, entry validation, you name it.

Sigurado ako magaling na kayo sa mga ito.

You can place them where you think is appropriate.

Here are my sample business logic. Hindi ko lahat nilagyan, just few to show you.

```java
/**
 *
 * @author Cody Scott "CodyScott"
 * @version 1
 *
 */
public class Vegetable {
    private String name;
    private String type;
    private String lasa;
    private boolean isFruit;
    //getter
    public String getName() {
        return name;
    }
    //setter
    public void setName(String name) {
        this.name = name;
    }
    //getter
    public String getType() {
        return type;
    }
    //setter
    public void setType(String type) {
        this.type = type;
    }
    //getter
    public String getLasa() {
        return lasa;
    }
    //setter
    public void setLasa(String lasa) {
        lasa.trim();
        if(lasa == null || lasa ==""){
            lasa = "Wala lang";
        } else {
            this.lasa = lasa;
        }
    }
    //getter
    public boolean isFruit() {
        return isFruit;
    }
    //setter
    public void setFruit(boolean isFruit) {
```

yan ang business logic ko for Vegetable.

:) lol..... 1 sample for you.

Of course, halimbawa lang yan.
pag wala silang nilagay na lasa ng vegetable, gagawing "Wala lang" ang default value.

This is where you need to think. Logic. Validation. etc.

You can also do it sa Fruit class, or Basket class... or MainApp class.

This is also where you need to be creative.

This is the MainApp.
Very very clean and easy to ready. I divide the processes into their own functions or methods.
(Of course, marami pang puwedeng i-improve, I could go a little further to simplify it but you guys get the point, hopefully)

```java
import java.util.ArrayList;

/**
 *
 * @author Cody Scott "CodyScott"
 * @version 1
 *
 */
public class MainApp {
    static Basket basket;
    static int fruitCount;
    static int vegetableCount;

    public static void main(String[] args) {

        initialize();                   //MainApp helper method (gawa gawa lang natin)

        createFruits();                 //MainApp helper method (gawa gawa lang natin)

        createVeggies();                //MainApp helper method (gawa gawa lang natin)

        basket.displayTotalItems();     //ready method from Basket class

        displayNamesOfItems();          //MainApp helper method (gawa gawa lang natin)
        displayCountOfEachItem();       //MainApp helper method (gawa gawa lang natin)
    }


    //helper method
    public static void initialize(){
        basket = new Basket();
        fruitCount=0;
        vegetableCount=0;
    }

    //helper method (taga gawa ng fruits)
    public static void createFruits(){
        Fruit apple = new Fruit();
        apple.setFruitName("Apple");
        apple.setFruitNumberOfSeeds(4);
        basket.addToBasket(apple);

        Fruit kaimito = new Fruit();
        kaimito.setFruitName("Apple");
        kaimito.setFruitNumberOfSeeds(8);
        basket.addToBasket(kaimito);

        Fruit rambutan = new Fruit();
        rambutan.setFruitName("Rambutan");
        rambutan.setFruitNumberOfSeeds(1);
        basket.addToBasket(rambutan);

        //adding 5 more kaimitos
        for(int i=0; i<5; i++){
            basket.addToBasket(kaimito);
        }
    }

    //helper method (taga gawa ng vegetables)
    public static void createVeggies(){
        Vegetable veggie1 = new Vegetable();
        veggie1.setName("Ampalaya");
        veggie1.setLasa("Mapait");
        veggie1.setFruit(true);
        basket.addToBasket(veggie1);

        Vegetable veggie2 = new Vegetable();
        veggie1.setName("Kangkong");
        veggie1.setLasa("Meh");
        veggie1.isFruit();
        basket.addToBasket(veggie2);

        //adding 3 more ampalaya
        for(int i=0; i<4; i++){
            basket.addToBasket(veggie1);
        }
    }

    //helper method
    public static void displayNamesOfItems(){
        Fruit aFruit = new Fruit();
        Vegetable aVeggie = new Vegetable();
        ArrayList<Object> list = basket.getBasket();
        for(Object item: list){
            if(item.getClass().isInstance(aFruit)){
                aFruit = (Fruit) item;
                fruitCount++;
                System.out.println(aFruit.getFruitName());
            } else if(item.getClass().isInstance(aVeggie)){
                aVeggie = (Vegetable) item;
                vegetableCount++;
                System.out.println(aVeggie.getName());
            }
        }
    }

    //helper method
    public static void displayCountOfEachItem(){
        System.out.println("Total Fruits: " + fruitCount);
        System.out.print("Total Vegetables: " + vegetableCount);
    }


}
```

readily available function from the Basket class

this the main processing of your program. The start and end of the whole processing.

6 lines of code

OK, guys. Goodluck sa inyong lahat. Keep coding and thanks for following this tutorial series. I hope this sparks a little bit of interest in you to learn more about OOP and Java.