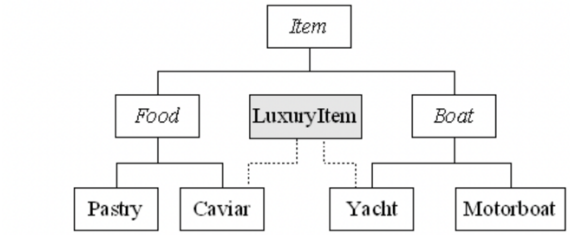
1. Introduction

The key point of this program is to embody LuxuryItem interface. LuxuryItem interface's function is to add luxury tax to yacht and caviar, so we should add function which adds luxury tax when calculating prices in yacht and caviar. Let's look at Luxury item interface first. Inheritance will be like this picture below.



2. Function description

-LuxuryItem Interface

```java
package HW6;

public interface LuxuryItem {
    public double calculateLuxuryTax();
}

```

This interface has one function. Named 'calculateLuxuryTax', this function calculates Luxury tax which occurred in Caviar and Yacht.

-Item Class

```java
package HW6;

import java.text.NumberFormat;
/**
* A generic item sold at a store
*/
abstract class Item {
    double cost;
    String name;
    /**
    * Constructs a new item with cost of zero and null name.
    */
    public Item( )
    {
        cost = 0.0;
        name = "";
    }
    /**
    * Constructs a new Item with the given name and cost.
    * @param name the item name.
    * @param cost the item's cost.
    */
    public Item( String name, double cost )
    {
        this.name = name;
        this.cost = cost;
    }
    /**
    * Set the cost of this Item.
    * @param cost the cost of the item.
    */
    public void setCost( double cost )
    {
        this.cost = cost;
    }
    /**
    * Get the cost of this Item
    * @return the cost of the item.
    */
    public double getCost()
    {
        return this.cost;
    }
    /**
    * Set the name of this item.
    * @param name the item name.
    */
    public void setName( String name )
    {
        this.name = name;
    }
    /**
    * Get the name of this Item.
    * @return the name of the item.
    */
    public String getName( )
    {
        return this.name;
    }
    /**
    * Return a string representation of this Item.
    * @return a string in format Name -- Cost: $cost.
    */
    public String toString( )
    {
        NumberFormat currency = NumberFormat.getCurrencyInstance();
        String result;
        result = name + " -- Cost: " + currency.format( cost );
        return result;
    }
    /**
    * Return a string that describes this item.
    * The default action is to return the item name
    */
    public String getDescription( )
    {
        return this.name;
    }
}
```

Item class has several methods. Most important methods is getDescription() methods. This methods will be overrided in inherited classes. Inherited classes are below.

-Food Class

```java
package HW6;

abstract class Food extends Item {
    int calories;
    /**
    * Construct a Food item with specified name, cost, and calorie count.
    * @param name the name of the food.
    * @param cost the cost of the food.
    * @param calories the number of calories per serving.
    */
    public Food( String name, double cost, int calories ) {
        super( name, cost );
        this.calories = calories;
    }
    /**
    * Get the number of calories per serving of this Food.
    * @return the number of calories per serving.
    */
    public int getCalories() {
        return this.calories;
    }
    /**
    * Set the number of calories per serving of this Food.
    * @param calories the calories per serving.
    */
    public void setCalories(int calories) {
        this.calories = calories;
    }
    /**
    * Return a string representation of this Food.
    * @return a string in format Name -- Cost: $cost Calories: calories.
    */
    public String toString() {
        return super.toString() + " Calories: " + calories;
    }
}
```

Food class is inherited by Item class, and have variable named calories. So we should make setter & getter about calories variable. Also override toString() methods that can return super class's information(item class) and this class's variable calories information.

-Pastry Class

```java
package HW6;

import java.text.NumberFormat;
import java.util.Locale;

public class Pastry extends Food {
    String flavor;

    public Pastry(String name, double cost, int calories, String flavor){
        super(name, cost, calories);
        this.flavor = flavor;
    }

    public String getFlavor(){
        return this.flavor;
    }

    public void setFlavor(String flavor){
        this.flavor = flavor;
    }

    public String getDescription( )
    {
        NumberFormat currency = NumberFormat.getCurrencyInstance(Locale.US);
        return flavor + " " + super.getDescription() + " " + currency.format(super.getCost());
    }
}

```

Pastry class is inherited by Food class. This class has flavor variable, so I made setter & getter about flavor variable. In getDescription() methods, overrides Item class's getDescription() methods, returns super class(food)'s description and also this pastry's information. And I formatted currency using NumberFormat.

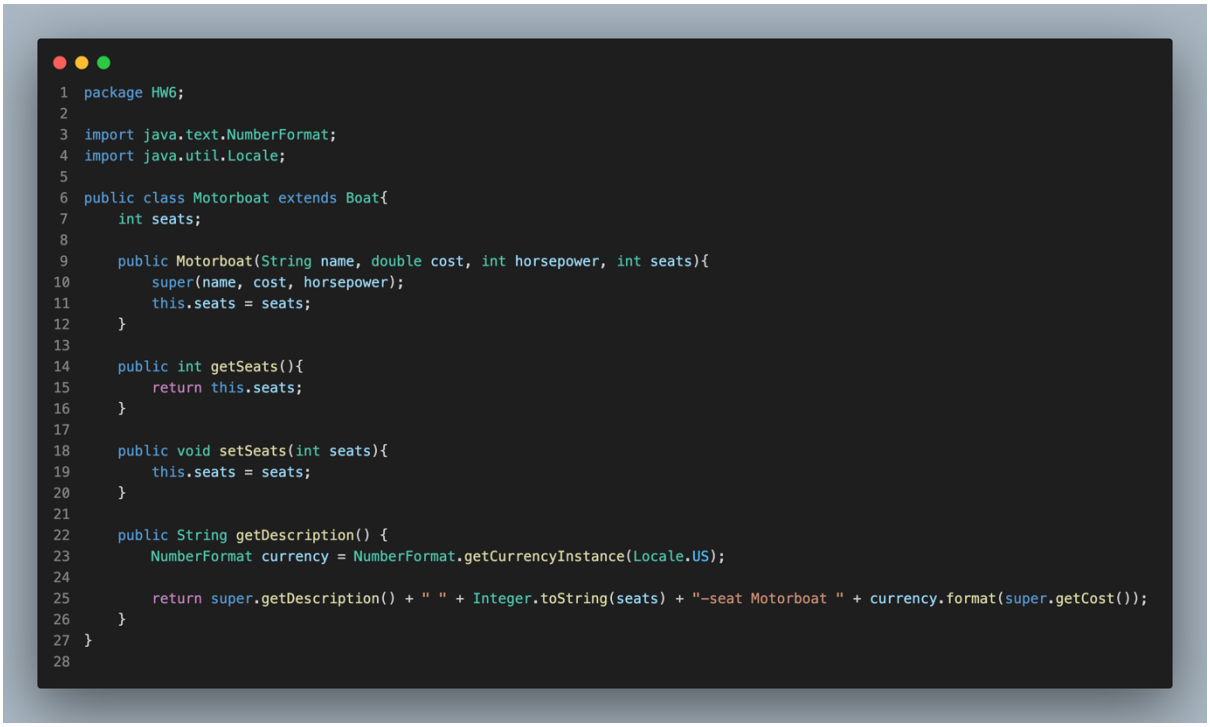
-Boat Class

```java
package HW6;

abstract class Boat extends Item {
    int horsepower;
    /**
    * Construct a Boat item with specified name, cost, and horsepower.
    * @param name the name of the food.
    * @param cost the cost of the food.
    * @param horsepower the horsepower of the boat's engine.
    */
    public Boat( String name, double cost, int horsepower ) {
        super( name, cost );
        this.horsepower = horsepower;
    }
    /**
    * Get the number of horsepower for this Boat.
    * @return the horsepower of this boat's engine.
    */
    public int getHorsepower() {
        return this.horsepower;
    }
    /**
    * Set the number of horsepower for this boat.
    * @param horsepower the boat engine's horsepower.
    */
    public void setHorsepower(int horsepower) {
        this.horsepower = horsepower;
    }
    /**
    * Return a string representation of this Food.
    * @return a string in format Name -- Cost: $cost Horsepower: horsepower.
    */
    public String toString() {
        return super.toString() + " Horsepower: " + horsepower;
    }
        }
```

Boat class is inherited by Item class, has one variable calls horsepower. So make getter& setter about horsepower and override toString() methods that can return its super class's information (item class) and this class's variable horsepower variable.

-Motorboat Class



```java
package HW6;

import java.text.NumberFormat;
import java.util.Locale;

public class Motorboat extends Boat{
    int seats;

    public Motorboat(String name, double cost, int horsepower, int seats){
        super(name, cost, horsepower);
        this.seats = seats;
    }

    public int getSeats(){
        return this.seats;
    }

    public void setSeats(int seats){
        this.seats = seats;
    }

    public String getDescription() {
        NumberFormat currency = NumberFormat.getCurrencyInstance(Locale.US);

        return super.getDescription() + " " + Integer.toString(seats) + "-seat Motorboat " + currency.format(super.getCost());
    }
}

```

Motor class is inherited by Boat class. It has one variable seats, so make getter & setter about seats variable. And override getDescription() methods to add description about this class. To make it clear, this Motorboat class is inherited by Boat class, and Boat class is inherited by Item class. So this getDescription() method has information about Item, Boat, and Motorboat.

-Yacht Class

```java
package HW6;

import java.text.NumberFormat;
import java.util.Locale;

public class Yacht extends Boat implements LuxuryItem{
    int cabins;

    public Yacht(String name, double cost, int horsepower, int cabins){
        super(name, cost, horsepower);
        this.cabins = cabins;
    }

    public double calculateLuxuryTax(){
        final double RATE = 0.1;
        return super.getCost() * RATE;
    }

    public String getDescription() {
        NumberFormat currency = NumberFormat.getCurrencyInstance(Locale.US);

        return super.getName() + " " + Integer.toString(cabins) + "-cabins yacht " +
        currency.format(super.getCost() + calculateLuxuryTax()) + " ** Luxury tax added **";
    }

    public int getCabins() {
        return this.cabins;
    }

    public void setCabins(int cabins){
        this.cabins = cabins;
    }
}

```

Yacht class is inherited by Boat class. And Most important, this is luxury item, so we should implement calculateLuxuryTax() to add taxes when we calculate price. (LuxuryItem interface is interface and only has methods declaration, so we should implement detail in class which uses LuxuryItem interface).

This class's getDescription() has information about Item, Boat, Yacht information and prices are calculated adding luxury taxes.

-Caviar Class

```java
package HW6;

import java.text.NumberFormat;
import java.util.Locale;

public class Caviar extends Food implements LuxuryItem {
    String origin;

    public Caviar(String name, double cost, int calories, String origin){
        super(name, cost, calories);
        this.origin = origin;
    }

    public double calculateLuxuryTax(){
        final double RATE = 0.15;
        return super.getCost() * RATE;
    }

    public String getOrigin(){
        return origin;
    }

    public void setOrigin(String origin){
        this.origin = origin;
    }

    public String getDescription(){
        NumberFormat currency = NumberFormat.getCurrencyInstance(Locale.US);

        return super.getName() + " caviar from " + getOrigin() + " " +
        currency.format(super.getCost() + calculateLuxuryTax()) + " ** Luxury tax added **";
    }
}

```

Caviar class is same with Yacht. This is also luxury item, so we should implement calculateLuxuryTax() method detail. Same as Yacht, this class's getDescription() method has information about Item, Food, Caviar.

3. Final Statement

This problem's most important thing is interface(LuxuryItem). This interface only has method declaration. So if we want to use this interface, we should declare this interface's methods clearly and detail.

And I used NumberFormat to print currency simply and plainly. This helps to calculate currency and can change currency easily. I used US dollar $.

4. Execution

```
junwoo@bagjun-uui-MacBookPro HW6 %  /usr/bin/env /Library/Java/JavaVirtualM
ion\ Support/Code/User/workspaceStorage/fb84ad29a432270a0ab6533430f7fd3a/r
Chocolate Croissant $1.99
Evinrude 5-seat Motorboat $3,475.00
oceano 12-cabins yacht $1,045,000.00 ** Luxury tax added **
Beluga caviar from Caspian Sea $655.50 ** Luxury tax added **
junwoo@bagjun-uui-MacBookPro HW6 %
```

We could see Yacht and Caviar classes are added luxury tax. So we can conclude we implemented interface correctly.