**Computer Language 2022**

**Assignment #4**

**Due: 8/May 23:59:59**

**Student ID: 21102052 Name: Lee Jeong-Yun**

**Carefully read the following descriptions.**

**Find CalcEx.java file and add implementations to meet the following requirements:**

**1) Add *Bootable* interface: this interface should have “public abstract void boot()” method and “default void shutdown()” method. The shutdown method must provide an implementation for printing out a shutdown message (“---Shutdown---“) to the console.**

**2) Add *Refundable* interface: this interface should have “public abstract void refund()” method.**

**3) Add *Device* abstract class: this class must have “private String vendor” field which will be set by Device(String name) constructor. Also, a getter method for this field (i.e., vendor) must be implemented here.   
Device class should have “abstract void printlLogo()” method and “public void turnOn()” which prints out “[vendor field] is on now…” to the console.**

**4) Add *Calculator* abstract class: this class extends Device class and implements Bootable and Refundable interfaces.   
The boot() method of Bootable interface must be implemented in this class. The boot() method prints out “<<< Calculator Bootup >>>” to the console.   
The refund() method of Refundable interface must be implemented in this class. The refund() method prints out “[vendor name](use getter method) + Refunded!” to the console.  
Also, this class provide the following public abstract methods: int add (int num1, int num2), int subtract(int num1, int num2), double average(int[] nums) that will be implemented by NormalCalculator and AbsCalculator.**

**5) *NormalCalculator* and *AbsCalculator* extend Calculator class, and must implement all abstract methods defined in Calculator, Device, and Bootable/Refundable interfaces. The difference between NormalCalculator and AbsCalculator is as follows:**

**- Add method takes two integer numbers and then prints the result of addition**

**- Subtract method takes two integer numbers and then prints the result of subtraction**

**- Average method takes an array of integer numbers and then prints the average**

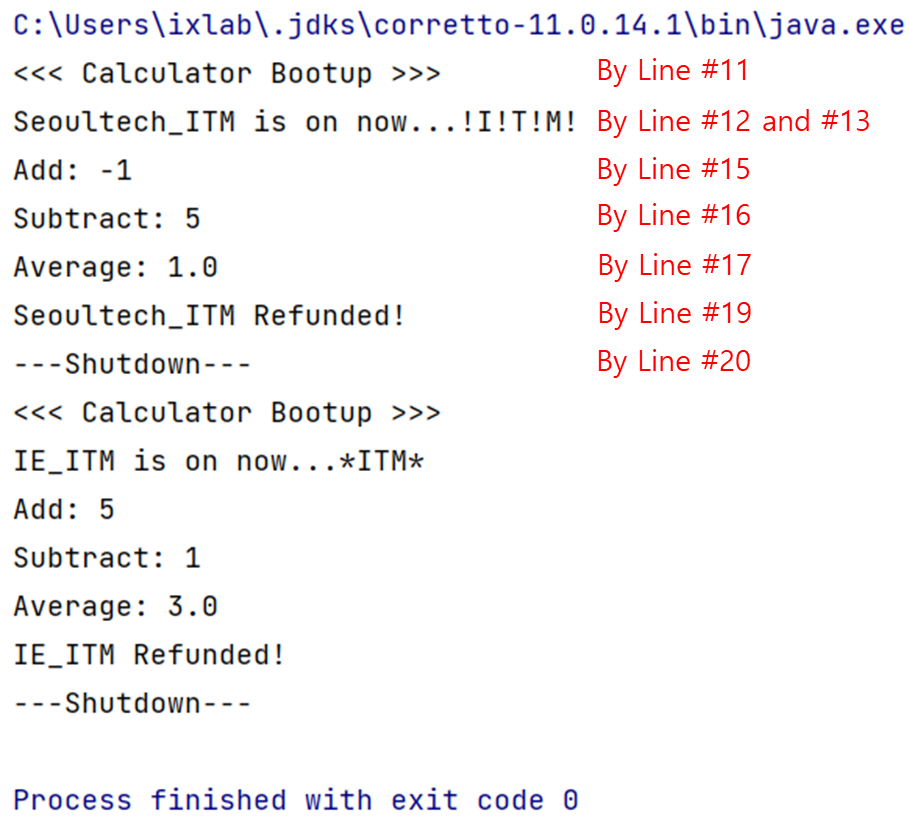
**- AbsCalculator does the same thing except that all the input parameters are converted to positive integer numbers (e.g,. myCalc.add(2, -3) will be treated like myCalc.add(2, 3))**

**- printLogo() method must be implemented by these classes. The printLogo() method of AbsCalculator and NormalCalculator prints “\*ITM\*" and “!I!T!M!”, respectively.**

**- Constructor which takes a single String parameter must be supported. This constructor should invoke a superclass’s constructor method to set the vendor field of Device class.**

**6) Line 1 ~23 cannot be modified.**

**Output)**

****

**Your code:**

public class CalcEx {  
public class CalcEx {  
  
 public static void main(String [] args) {  
  
 Assignment.HW04.Calculator[] calcs = {  
 new Assignment.HW04.NormalCalculator("Seoultech\_ITM"),  
 new Assignment.HW04.AbsCalculator("IE\_ITM")  
 };  
  
 for(Assignment.HW04.Calculator myCalc : calcs) {  
 myCalc.boot();  
 myCalc.turnOn();  
 myCalc.printLogo();  
  
 System.out.println("Add: "+myCalc.add(2, -3));  
 System.out.println("Subtract: "+myCalc.subtract(2, -3));  
 System.out.println("Average: "+myCalc.average(new int[]{2, -3, 4}));  
  
 myCalc.refund();  
 myCalc.shutdown();  
 }  
 }  
}  
  
interface Bootable{  
 public abstract void boot();  
 default void shutdown() {  
 System.out.println("---Shutdown---");  
 }  
}  
  
interface Refundable{  
 public abstract void refund();  
}  
  
abstract class Device {  
 private String vendor;  
 public Device(String name) {  
 vendor = name;  
 }  
 public String vendorGetter() {  
 return vendor;  
 }  
 abstract void printLogo();  
 public void turnOn() {  
 System.out.print(vendorGetter()+" is on now...");  
 }  
}  
  
abstract class Calculator extends Assignment.HW04.Device implements Assignment.HW04.Bootable, Assignment.HW04.Refundable {  
 public Calculator(String name) {  
 super(name);  
 }  
 public void boot() {  
 System.out.println("<<< Calculator Bootup >>>");  
 }  
 public void refund() {  
 System.out.println(vendorGetter()+" Refunded!");  
 }  
 public abstract int add(int num1, int num2);  
 public abstract int subtract(int num1, int num2);  
 public abstract double average(int[] nums);  
}  
  
class NormalCalculator extends Assignment.HW04.Calculator {  
 NormalCalculator(String name) {  
 super(name);  
 }  
 public void printLogo() {  
 System.out.println("!I!T!M!");  
 }  
 public int add(int a, int b) {  
 return a+b;  
 }  
 public int subtract(int a, int b) {  
 int result = a-b;  
 if (result < 0){  
 result = -result;  
 }  
 return result;  
 }  
 public double average(int[] a) {  
 int sum = 0;  
 for (int i : a) {  
 sum = sum + i;  
 }  
 int avg = sum/a.length;  
 return avg;  
 }  
}  
  
class AbsCalculator extends Assignment.HW04.Calculator {  
  
 AbsCalculator(String name) {  
 super(name);  
 }  
 public void printLogo() {  
 System.out.println("\*ITM\*");  
 }  
 public int add(int a, int b) {  
 if (a < 0){  
 a = -a;  
 }  
 if (b < 0){  
 b = -b;  
 }  
 return a+b;  
 }  
 public int subtract(int a, int b) {  
 if (a < 0){  
 a = -a;  
 }  
 if (b < 0){  
 b = -b;  
 }  
 int result = a-b;  
 if (result < 0){  
 result = -result;  
 }  
 return result;  
 }  
 public double average(int[] a) {  
 int sum = 0;  
 for (int i : a) {  
 if (i < 0){  
 i = -i;  
 }  
 sum = sum + i;  
 }  
 int result = sum/a.length;  
 return result;  
 }  
}

**Your result (screenshot):**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**Your explanation on the code:**

**Interface Bootable and Refundable have abstract method boot() and refund(). Bootable interface has also shutdown() method.**

**Device is abstract class. It has private field called ‘vendor’ and constructor which can get single String parameter. And the parameter is assigned to vendor. Then, I made getter method return private field, vendor. printLogo() method is abstract, and turnOn() method is just method in Device class.**

**Calculator abstract class extends Device class and implements Bootable and Refundable interface. Because of this, Calculator should redefine the interfaces’ methods. It has its own additional abstract methods such as add, subtract, and average. The constructor of the class uses super keyword to pass the parameter to its superclass.**

**NormalCalculator and AbsCalculator are default classes. They extend Calculator. So, they must redefine abstract methods like printLogo(), add(), subtract(), and average(). Their constructors have super keyword also, to pass the parameter to their parent class, Calculator.**

**add() method returns the sum of parameters, a and b. subtract() method returns the difference of the parameters, a and b. If a-b is negative, then change the sign of the number to express positive number. average() method has array parameter, so using for statement to get each elements of the array and add the sum variable to get total sum of elements. And then, sum is divided by array’s length. The result type is int, so the return value could be not exact average. These logics are common to both classes.**

**However, in AbsCalculator, it requires additional condition. All parameters in operation should be positive even if the input is negative. So, I added sign converter if the parameters are smaller than 0.**