**Exercise 1:**

Give the output of the following program.

|  |
| --- |
| **public** **abstract** **class** Vehicle { **protected** String brand; **protected** **double** price;  **public** Vehicle() { brand = "Unknown"; price = 50.0; System.*out*.println(" .... Brand : " + brand + " --- Price : " + price); } **public** Vehicle(String b, **double** p) { brand = b; price = p; System.*out*.println(" .... Brand : " + brand + " --- Price : " + price); } **public** **void** show() { System.*out*.println(" .... Brand : " + brand + " --- Price : " + price); } } |
| **public** **class** Bus **extends** Vehicle { **private** String name; **private** **int** nbOfSeats;  **public** Bus(){ name = "Hafeela"; nbOfSeats =11; show(); } **public** Bus(String s, String b, **double** p, **int** n) { name = s; brand = b; price = p; nbOfSeats = n;  show(); } **public** **void** show() { System.*out*.println(" \*\*\*\* Name : " + name + " .... Nb of Seats : " + nbOfSeats);  **super**.show(); } **public** **void** addPassangers(**int** nb) **throws** Exception{ **if** (nb <= 0) **throw** **new** Exception ("Unaccepted parameter value"); **if** (nb > nbOfSeats) **throw** **new** Exception ("Parameter value exceeds available seats");  nbOfSeats -= nb ; show(); }} |
| **public** **class** Testing { **public** **static** **void** main(String[] args) { Bus m1 = **new** Bus(); System.*out*.println("+++++++++++++++++++++++++++++++++++++++++++"); Bus m2 = **new** Bus("m2", "Mercedes", 70.0, 5); System.*out*.println("==========================================="); **try** { m1.addPassangers(10); System.*out*.println("--------------------------------"); m2.addPassangers(10); } **catch**(Exception e) {  System.*out*.println (e.getMessage());  } }} |

**Exercise 2:**



***MobileDevice*** class***:***

* + Attributes:
		- ***brand***: the brand of the mobile device.
		- ***speed:*** the speed of the mobile device.
		- ***storage***: the storage capacity of the mobile device.
		- ***price***: the price of the mobile device.
	+ Methods:
* ***MobileDevice(brand: string, storage: int, speed: double)***: constructor
* ***calculatePrice(* ):** this method calculates and returns the price of the mobile device. The price is calculated as follows:
	+ ***For Smart Phone****: price = 1300 + number of sim cards \* 150*
	+ ***For Tablet:*** *price = 1300 \* storage+ screen size \* 50*
* ***getBrand():***this method returns the brand of the mobile device.

***SmartPhone*** class

* + Attributes:
		- ***nbSim***: the number of SIM cards.
	+ Methods:
* ***SmartPhone (brand: string, storage: int, speed: double, nbSim: int)***: constructor.
* ***getNbSim():***this method returns the number of sim cards supported by the smart phone.

***Tablet*** class:

* + Attributes:
		- ***screenSize***: the screen size of the tablet.
	+ Methods:
* ***Tablet (brand: string, storage: int, speed: double, screenSize: int)***: constructor.

***Company*** class***:***

* + Attributes:
		- ***name***: the name of the company.
	+ Methods:
* ***Company(name: string, size: int)***: constructor
* ***addMobile(m:MobileDevice*):** this method adds the mobile device ***m*** to the company. It returns the string “Successfully added” if the mobile device ***m*** is added. Otherwise, it returns “Cannot be added”.
* ***countMobileDevices(b:string):*** this method counts and returns the number of mobile devices having the brand ***b***.
* ***averagePricesOfSmartPhones():*** this method calculates and returns the average price of the smart phones only.
* ***averagePricesOfSmartPhones(nbS:int, b:string):*** this method calculates and returns the average price of the smart phones of the brand ***b*** and having ***nbSim*** greater than ***nbS***.
* ***saveSmartPhones(filename:string, nbS:int):*** this method saves the smart phones having ***nbSim*** greater or equal to ***nbS*** into the object file “filename”, and returns the number of saved objects.

**QUESTION**: Translate into Java code the class ***MobileDevice***, the class ***SmartPhone,*** and the class ***Company.***

Answer Question 1:

.... Brand : Unknown --- Price : 50.0

 \*\*\*\* Name : Hafeela .... Nb of Seats : 11

 .... Brand : Unknown --- Price : 50.0

+++++++++++++++++++++++++++++++++++++++++++

 .... Brand : Unknown --- Price : 50.0

 \*\*\*\* Name : m2 .... Nb of Seats : 5

 .... Brand : Mercedes --- Price : 70.0

===========================================

 \*\*\*\* Name : Hafeela .... Nb of Seats : 1

 .... Brand : Unknown --- Price : 50.0

--------------------------------

Parameter value exceeds available seats

Answer Question2:

**public** **abstract** **class** MobileDevice { **--------------- 1**

 **private** String brand;

 **private** **double** speed;

 **protected** **int** storage;

 **protected** **double** price;

 **public** MobileDevice(String b, **int** st, **double** sp) { **--------------- 1**

 brand = b;

 storage = st;

 speed = sp;

 price = 0.0;

 }

 **public** MobileDevice(MobileDevice m) { **--------------- 2**

 brand = m.brand;

 storage = m.storage;

 speed = m.speed;

 price = m.price;

 }

 **public** **abstract** **double** calculatePrice();**--------------- 1**

 **public** String getBrand() { **--------------- 1**

 **return** brand;

 }

}

**public** **class** SmartPhone **extends** MobileDevice { **--------------- 1**

 **private** **int** nbSim;

 **public** SmartPhone(String b, **int** st, **double** sp, **int** nbS) {

 **super**(b, st, sp); **--------------- 1**

 nbSim = nbS;

 }

 **public** SmartPhone(SmartPhone s) {

 **super**(s); **--------------- 1**

 nbSim = s.nbSim;

 }

 **public** **int** getNbSim() { **--------------- 1**

 **return** nbSim;

 }

 **public** **double** calculatePrice() { **--------------- 1**

 price = 1300 + nbSim \* 150.0;

 **return** price;

 }

}

**public** **class** Company {

 **private** String name;

 **private** MobileDevice arrMob[];

 **private** **int** nbMD;

 **public** Company(String name, **int** size) {

 **this**.name = name;

 arrMob = **new** MobileDevice[size]; **--------------- 1**

 nbMD = 0; **--------------- 1**

 }

 **public** String addMobile(MobileDevice m) {

 **if** (nbMD < arrMob.length) { **--------------- 1**

 **if** (m **instanceof** SmartPhone) **--------------- 1**

 arrMob[nbMD] = **new** SmartPhone( (SmartPhone) m); **----1**

 **else**

 arrMob[nbMD] = **new** Tablet( (Tablet) m); **----1**

 nbMD ++; **----1**

 **return** "Sucessfully added"; **----0.5**

 }

 **else**

 **return** "Can not be added"; **----0.5**

 }

 **public** **int** countMobileDevices (String b) {

 **int** count = 0; **--------------- 0.5**

 **for** (**int** i = 0; i < nbMD; i++) { **--------------- 1**

 **if** (arrMob[i].getBrand().equals(b)) **--------------- 1**

 count ++; **--------------- 1**

 }

 **return** count; **--------------- 0.5**

 }

 **public** **double** averagePricesOfSmartPhones () {

 **int** count = 0; **--------------- 0.5**

 **double** sum = 0.0; **--------------- 0.5**

 **double** avg = 0.0;

 **for** (**int** i = 0; i < nbMD; i++) { **--------------- 1**

 **if** (arrMob[i] **instanceof** SmartPhone ) { **--------------- 1**

 count ++; **--------------- 1**

 sum += arrMob[i].calculatePrice(); **--------------- 1**

 }

 }

 **if** (count > 0)

 avg = sum / count; **--------------- 0.5**

 **return** avg; **--------------- 0.5**

 }

**public** **double** averagePricesOfSmartPhones (**int** nbS, String b) {

 **int** count = 0; **--------------- 0.5**

 **double** sum = 0.0; **--------------- 0.5**

 **double** avg = 0.0;

 SmartPhone s;

 **for** (**int** i = 0; i < nbMD; i++) { **--------------- 1**

 **if** (arrMob[i] **instanceof** SmartPhone ) { **--------------- 1**

 s = (SmartPhone) arrMob[i]; **--------------- 1**

 **if** ( arrMob[i].getBrand().equals(b) && **---------- 1**

s.getNbSim() > nbS) { **--------------- 1**

 count ++; **--------------- 1**

 sum += arrMob[i].calculatePrice(); **------- 1**

 }

 }

 }

 **if** (count > 0)

 avg = sum / count; **--------------- 0.5**

 **return** avg; **--------------- 0.5**

 }

 **public** **int** saveSmartPhones(String fileName, **int** nbS) **throws** IOException {

 File f = **new** File (fileName); **--------------- 0.5**

 FileOutputStream fo = **new** FileOutputStream(f); **--------------- 1**

 ObjectOutputStream objF = **new** ObjectOutputStream(fo); **--------- 1**

 **int** count = 0; **--------------- 0.5**

 SmartPhone s;

 **for** (**int** i=0; i < nbMD; i++) { **--------------- 1**

 **if** (arrMob[i] **instanceof** SmartPhone) { **--------------- 1**

 s = (SmartPhone) arrMob[i]; **--------------- 1**

 **if** (s.getNbSim() > nbS) { **--------------- 1**

 objF.writeObject(arrMob[i]); **--------------- 1**

 count ++; **--------------- 1**

 }

 }

 }

 objF.close(); **--------------- 0.5**

 **return** count; **--------------- 0.5**

 }

}