WORKSHOP 02

Exercise 1:

1. Design and code a class named **Fan** that holds information about a fan and class named **ColorFan** which is derived from **Fan** (i.e. Fan is super class and ColorFun is sub class).

Fan
-code:String -price:double
+Fan () +Fan (code:String, price:double) +getCode():String +getPrice():double +**toString()**:String

Where:

- getCode(): String – return code.
- getPrice(): double – return price.
- setPrice(price:double): void – set this.price = price.
- toString():String – return the string of format:

code price

ColorFan
-series:int
+ColorFan () +ColorFan (code:String, price:double, series:int) +getPrice():double +**toString()**:String

Where:

- getPrice(): double – check if series<300 then return new price = original price + inc, where inc = 10% of the original price, otherwise return original price.
- toString():String – return the string of format:

code price series

2. Build Test class contains main function including:

- Properties:

+ **n** used to store the length of list 0 <n <= 100.

+ **k** is used to store the number of **ColorFan** currently on the list.

+ **lex** is used to store the list of **ColorFan**.

+ **s** is a Scanner variable used in data entry.

- Create methods:

+ **Input()** to enter an **ColorFan** on the list.

+ **Output()** to display an the list of **ColorFan**.

+ **Search()** to search for an **ColorFan** by Name.

- Write the main() method that runs the above methods:

+ enter a list of n **ColorFan**;

+ display the list of **ColorFan**;

+ enter the Name of the **ColorFan** and display the information of this **ColorFan** who has that Name.

Exercise 2:

1. Write a class named **Car** that holds information about a Car and class named **VNCar** which is derived from **Car** (i.e. Car is super class and VNCar is sub class).

Car
-name:String -price:double
+Car() +Car(name:String, price:double) +getName():String +getPrice():double +**toString()**:String

Where:

- getName(): String – return name.
- getPrice(): double – return price.
- setPrice(price:double): void – update price.
- toString():String – return the string of format:

name price

VNCar
-series:int
+VNCar () +VNCar (name:String, price:double, series:int) +getSalePrice():double +**toString()**:String

Where:

- getSalePrice(): double – return the value price1=price+inc, where inc = 10% of price if series<300, = 0 otherwise.
- toString():String – return the string of format:

name price series

2. Build Test class contains main function including:

- Properties:

+ **n** used to store the length of list 0 <n <= 100.

+ **k** is used to store the number of **VNCar** currently on the list.

+ **lex** is used to store the list of **VNCar**.

+ **s** is a Scanner variable used in data entry.

- Create methods:

+ **Input()** to enter an **VNCar** on the list.

+ **Output()** to display an the list of **VNCar**.

+ **Search()** to search for an **VNCar** by Name.

- Write the main() method that runs the above methods:

+ enter a list of n **VNCar**;

+ display the list of **VNCar**;

+ enter the Name of the **VNCar** and display the information of this **VNCar** which has that Name.

Exercise 3

1) Create a abstract class named **Person** that contains:

- Attributes: ID, Name, Gender

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, abstract method for calculating Result.

2) Create child classe of **Person** named Candidate that contains:

- Attributes: ClassID, Mark

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, method for calculating Result, given if Mark > 8.5 then result is "Pass", otherwise "Slip".

3) Build Test class to create list of Candidates including:

- Properties:

+ **n** used to store the length of list 0 <n <= 100.

+ **k** is used to store the number of Candidates currently on the list.

+ **lex** is used to store the list of Candidate.

+ **s** is a Scanner variable used in data entry.

- Create methods:

+ **Input()** to enter an Candidate on the list.

+ **Output()** to display an the list of Candidates.

+ **Search()** to search for an Candidate by Name.

- Write the main() method that runs the above methods:

+ enter a list of n Candidates;

+ display the list of Candidate;

+ enter the Name of the Candidate and display the information of this Candidate who has that Name.

Exercise 4

1) Create a abstract class named **Person** that contains:

- Attributes: ID, Name, Gender

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, abstract method for calculating Result.

2) Create child classe of **Person** named Student that contains:

- Attributes: ClassID, Mark

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, method for calculating Result, given if Mark > 5 then result is "Pass", otherwise "Slip".

3) Build Test class to create list of Students including:

- Properties:

+ **n** used to store the length of list 0 <n <= 100.

+ **k** is used to store the number of Students currently on the list.

+ **lex** is used to store the list of Student.

+ **s** is a Scanner variable used in data entry.

- Create methods:

+ **Input()** to enter an Student on the list.

+ **Output()** to display an the list of Students.

+ **Search()** to search for an Student by ClassID.

- Write the main() method that runs the above methods:

+ enter a list of n Students;

+ display the list of Student;

+ enter the ClassID of the Student and display the information of this Student who has that ClassID.

Exercise 5

1) Create a abstract class named **Person** that contains:

- Attributes: ID, Name, Gender

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, abstract method for calculating Salary.

2) Create child classe of **Person** named **Teacher** that contains:

- Attributes: Address, Grading

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, method for calculating Salary, given Salary=1.500.000* Grading.

3) Build Test class to create list of Teachers including:

- Properties:

+ **n** used to store the length of list 0 <n <= 100.

+ **k** is used to store the number of Teachers currently on the list.

+ **lex** is used to store the list of Teacher.

+ **s** is a Scanner variable used in data entry.

- Create methods:

+ **Input()** to enter an Teacher on the list.

+ **Output()** to display an the list of Teachers.

+ **Search()** to search for an Teacher by Address.

- Write the main() method that runs the above methods:

+ enter a list of n Teachers;

+ display the list of Teacher;

+ enter the Address of the Teacher and display the information of this Teacher who has that Address.

Exercise 6

1) Create a abstract class named **Product** that contains:

- Attributes: ID, Name, Color

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, abstract method for calculating discount.

2) Create child classe of **Product** named Car that contains:

- Attributes: Year, Price.

- Methods: setters/ getters for attributes, explicit constructors, explicit constructors, method for calculating discount, given if year <2007 then discount=10%* Price.

3) Build Test class to create list of Cars including:

- Properties:

+ **n** used to store the length of list 0 <n <= 100.

+ **k** is used to store the number of Cars currently on the list.

+ **lex** is used to store the list of Car.

+ **s** is a Scanner variable used in data entry.

- Create methods:

+ **Input()** to enter an Car on the list.

+ **Output()** to display an the list of Cars.

+ **Search()** to search for an Car by Color.

- Write the main() method that runs the above methods:

+ enter a list of n Cars;

+ display the list of Car;

+ enter the Color of the Car and display the information of this Car which has been made by that Color.