**Workshop 2**

**Q1.** **Vehicle** Management System, including: car, motorbike, truck. Each of the Vehicle has attributes of VIN (Vehicle Identification Number), manufacturer, manufacture year, cost, color. Moreover, cars have types of engine, number of seats. Motorbikes have power and trucks have load.

Code a class named **Car**, **Motor**, **Truck** are directly derived from base class **Vehicle** object.

Write a program with the following functions:

1. Input data of vehicles (validation data if necessarily)
2. To view the details of the vehicles (3 vehicles in one table or one for each table), to count the number of vehicles in the following table.
3. Searching by manufacturer, manufacture year, cost, color, types of engine, seats, … (exactly or approximately)
4. Searching from .. to…( manufacture year, cost)
5. Sorting by manufacturer, manufacture year, cost, color, types of engine, seats, …
6. To calculate statistics

**Q2.**

Library Management, including Book, Magazine, Newspaper. Each of the Document has attributes of code, publisher and circulation (tổng số phát hành). Moreover, Books have *author, title and the number of pages*. Magazines have *issue number, monthly release*. Newspapers have *Release date*.

Code a class named **Book**, **Magazine**, **Newspaper** are directly derived from base class **Document** object.

Write a program with the following functions:

1. Input data of documents (validation data if necessarily)
2. To view the details of all of documents (3 documents in one table or one for each table), to count the number of documents in the following table.
3. Remove, update for a document (enter a code of document)
4. Searching by code, publisher, circulation, author, title and the number of pages … (exactly or approximately)
5. Searching from .. to…( circulation, the number of pages, monthly release, Release date)
6. Sorting by code, publisher, circulation, author, title and the number of pages …
7. To calculate statistics

**Q3.**

One IT center requies full-time and part-time teachers. Full-time teacher information has teacher code, full name, email, Professor (Học hàm) (non, Prof. (Professor), Assoc. Prof. (Asscociate Professor), Assist. Prof. (Assistant Professor)), University degree (học vị) (**Bachelor, Master, Post Doctor**), address, phone, number of teaching hours per month, basic pay (lương cơ bản), pay rate (hệ số lương) (**Bachelor: 1, Master: 1.1, Post Doctor: 1.2 and** Prof. : +0.3, Assoc. Prof. +0.2, Assist. Prof. +0.1), Norm hour (số giờ định mức), overtime hour (50.000 vnd/1h).

Part-time teacher information has teacher code, full name, email, Professor (non, Prof. (Professor), Assoc. Prof. (Asscociate Professor), Assist. Prof. (Assistant Professor)), University degree (**Bachelor, Master, Post Doctor**), address, phone, number of teaching hours per month, pay rate (**Bachelor: 1, Master: 1.1, Post Doctor: 1.2 and** Prof. : +0.3, Assoc. Prof. +0.2, Assist. Prof. +0.1), office, (200.000 vnd/1h).

Write a program with the following functions:

1. Enter the information for full-time and part-time teacher (validation data if necessarily)
2. Display all teacher information (in one table or one for each table), to count the number of teachers in the following table.
3. Remove, update a teacher (enter code of teacher)
4. Searching by code, full name, email, Professor, Assoc. Prof., University degree, address, phone, number of teaching hours per month, **basic pay**, pay rate … (exactly or approximately)
5. Searching from .. to…( number of teaching hours per month, pay rate, …)
6. Sorting by code, full name, email, Professor, Assoc. Prof., University degree, address, phone, number of teaching hours per month, **basic pay**, pay rate …
7. To calculate statistics

**Q4. We have the design for Phone number:**

|  |
| --- |
| **PhoneNumber** |
| * int area * String number |
| + PhoneNumber()  + PhoneNumber(int a, String n);  + **void display();** |

|  |
| --- |
| **IntPhoneNumber** |
| * String countryCode |
| + IntPhoneNumber ()  +IntPhoneNumber (String cc, int a, String n);  + **void display();** |

**Implement the above classes and make your own main program to test all the above methods. The output of your program something look like:**

|  |
| --- |
| Enter list of phone numbers  -----------------------------------------  Type of phone number ? (1 – local phone, 2 – Inter phone number, 0 - exit): 1  Enter area code: 111  Enter number: 111111  Type of phone number ? (1 – local phone, 2 – Inter phone number, 0 - exit): 2  Enter country code: 22  Enter area code: 222  Enter number: 222222  Type of phone number ? (1 – local phone, 2 – Inter phone number, 0 - exit): 1  Enter area code: 333  Enter number: 333333  Type of phone number ? (1 – local phone, 2 – Inter phone number, 0 - exit): 2  Enter country code: 44  Enter area code: 444  Enter number: 444444  Type of phone number ? (1 – local phone, 2 – Inter phone number, 0 - exit): 0  List of phone number:  ----------------------------------------  111 – 111111  22 – 222 – 222222  333 – 333333  44 - 444 – 444444 |

**Note:** You can using an array of base class PhoneNumber object

**PhoneNumber phonelist[] = new PhoneNumber[1000]; or**

**ArrayList< PhoneNumber > phonelist = new ArrayList< PhoneNumber >();**

**Q5**

We have the design for Staff:

|  |
| --- |
| **Person** |
| - String name |
| + Person()  + Person(String n)  + **void display();**  **+ double getSalary();** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Officer**   |  | | --- | | **Worker** | | - double hrs  - final double RATE = 5.5 | | + Worker()  +Worker(String n, double hrs)  + **void display();**  **+ double getSalary();** | |
| - double bSalary |
| + Officer ()  +Officer (String n, double s)  + **void display();**  **+ double getSalary();** |

- Person. getSalary() is an abstract method

- Worker and Officer are subclasses of Person and Worker.salary = hrs \* rate

Implement the above classes and make your own main program to test all the above methods

Add the below code to your main function and run

**Q6.** Create a class named DictionaryWord as:

|  |
| --- |
| DictionaryWord |
| - word: String                                                              - meanings: String |
| + DictionaryWord (String word, String meanings) + getWord(): String + setWord (String word): void + getMeanings(): String + setMeanings(String meanings): void |

Write a program with the following requirements:  
Creates 8 DictionaryWord objects with:

* Word and meanings as the table:

|  |  |
| --- | --- |
| ***Word*** | ***Meanings*** |
| bank robber | Steals money from a bank |
| Burglar | Breaks into a home to steal things |
| Forger | Makes an illegal copy of something |
| Hacker | Breaks into a computer system |
| Hijacker | Takes control of an airplane |
| Kidnapper | Holds someone for ransom money |
| Mugger | Attacks and steals money from someone |
| Murderer | Kills another person |

* Ensure that there is no duplicate DictionaryWord objects (02 DictionaryWord objects a and b are equal when a.word=b.word).

Displays all DictionaryWord in ascending order of word with the format as:

<<no>.<<word>>  
<<meanings>>

<<no>.<<word>>  
<<meanings>>

Where: <<no>>=1,2…

Hint:

* class DictionaryWord implements Comparable to order 2 DictionaryWord objects.
* override equals(..) method to compare 2 DictionaryWord objects.
* override toString()
* use Set to ensure no duplicate.
* use support class Collections to sort DictionaryWord objects.

**Q7.**

Create an interface for 2D geometry named Geometric2D includes the following methods:

* Calculate perimeter
* Calculate area
* Calculate the number of vertices
* Calculate the number of edges

Create an interface for 3D geometry named Geometric3D adds more the following methods:

* Calculete volume
* Calculate area around

Implement specific geometries include: square, rectangular, triangle, rhombus, parallelogram, pentagon, circle, sphere, boxy, cylinder….