SCIT

School of Computing and Information Technology
Faculty of Engineering & Information Sciences

CSIT121
Object Oriented Design and Programming
Assignment 1

Objectives:

Practice java programming with classes and objects, constructors, copy constructors, enum type, array, `ArrayList` (generic version), overloaded methods, passing by reference and etc.

Task (4 marks)

In mathematics, a set is a collection of *distinct* elements and elements in a set are not in order. Here are some examples of sets:

 (a) A set of integers, e.g. integerSet = {3, 1, 4, 2}
 (b) A set of fruits, e.g. fruitSet = {apple, orange, papaya}
 (c) A set of characters, e.g. charSet = {'A', 'm', '&'}

We always enclose elements of sets inside a pair of { }.

Here are some other properties on sets:

(1) A set can be empty, i.e. no element. We call it empty set. In mathematics, we have a special symbol to denote empty set. Convenient to our design later, we will use { } to denote an empty set.

(2) When checking an element is inside a set. We call it "belong to".

(3) If a set contains in another set, we call it subset. For example, {1, 2, 3} is a subset of {2, 3, 4, 6, 1}. Therefore empty set is a subset of every set.

(4) The *cardinal number* of a set is the number of elements in a set.

(5) The union of two sets A and B are all the elements belong to A and B, minus the duplications. For example, if A = {1, 2, 3} and B = {2, 3, 4, 5}, the union is {1, 2, 3, 4, 5}.

(6) The intersection of two sets A and B are the common elements of A and B. Using the example quoted in (5), the intersection is {2, 3}.

(7) The equality of two sets A and B are all the elements of A are in B and all the elements of B are in A. Or alternatively, A is the subset of B and vice versa.

(8) The difference of two sets A and B, for example A − B, is those elements in B should not be in A. For example, if A = {1, 2, 3} and B = {2, 3, 4, 5}, A − B = {1} and B − A = {4, 5}.

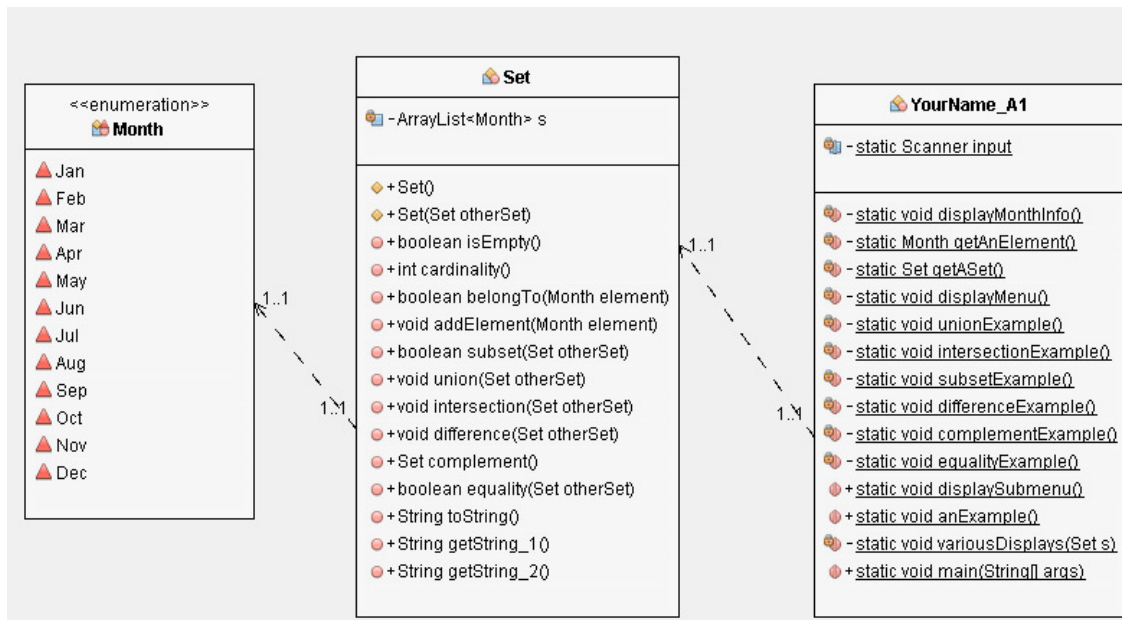We have all the required properties for our task.

In this assignment, our **universal set** is the 12 enumeration constants of calendar months. They are

Jan, Feb, Mar, …, Dec

Data validation is not necessary in this design, when "I" want to add in a month, you can assume that this is a valid month in the enumeration constants; i.e. we have restricted ourselves in a subset of the above 12 months.

The best way to test your design is to develop an educational system to teach some basic set theory. All the sets used in your design should be *randomly generated* by the system i.e. the sizes (also known as cardinal number, from 0 to 12 elements) and the elements.

Let us explore the following UML diagram for the whole task:

Month (<<enumeration>>): Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

Set
- ArrayList<Month> s
- +Set()
- +Set(Set otherSet)
- +boolean isEmpty()
- +int cardinality()
- +boolean belongTo(Month element)
- +void addElement(Month element)
- +boolean subset(Set otherSet)
- +void union(Set otherSet)
- +void intersection(Set otherSet)
- +void difference(Set otherSet)
- +Set complement()
- +boolean equality(Set otherSet)
- +String toString()
- +String getString_1()
- +String getString_2()

YourName_A1
- static Scanner input
- static void displayMonthInfo()
- static Month getAnElement()
- static Set getASet()
- static void displayMenu()
- static void unionExample()
- static void intersectionExample()
- static void subsetExample()
- static void differenceExample()
- static void complementExample()
- static void equalityExample()
- +static void displaySubmenu()
- +static void anExample()
- static void variousDisplays(Set s)
- +static void main(String[] args)

Let us look at each of the classes

(a) Enumeration class `Month`

This is the universal set that a subset is constructed. Each enum constant has two descriptions: an integer to denote the month, for example 1 for Jan; and a string to spell out the month in full, for example, January for Jan

(b) Class Set

We use an array list to represent a set, which is an instance variable defined inside the class. The normal set operations: belong to, contains, union, intersection, complement, difference, subset, equality are some of the set's operations. The toString method returns a String of enum constants enclosed between braces; getString_1 returns its equivalent in integers for elements of set and getString_2 returns its equivalent in string objects for elements of set.

(c) Main class

We now ready to present the whole system. You are required to design an educational system to teach basic set theory. We propose the following interactions for your system:

When you execute your program, the system will display the info for the universal set (a call to `displayMonthInfo` method, display once only) and followed by a menu (the display of menu is repeated after an operation)

```
Universal set info

        Jan  01        January
        Feb  02       February
        Mar  03          March
        Apr  04          April
        May  05            May
        Jun  06           June
        Jul  07           July
        Aug  08         August
        Sep  09      September
        Oct  10        October
        Nov  11       November
        Dec  12       December
```

```
Welcome to SIM Set Theory lesson

0: Properties of set
1. Union example
2. Intersection example
3. Subset example
4. Difference example
5. Complement example
6. Sets equality example
9. Quit

Your option: 0
```

When you enter the option 0, you will see the following interactions:

```
Your option: 0

Here is an example of set
        A = {Mar, Jun, Sep}
        All elements in set are distinct and random order

Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Various displays
        9: Quit

Enter your option: 1
```

A set is generated and is displayed. You can now try a few simple set operations, add an element, belong to operation, display the cardinal number and various ways of displays. Note that the sub-menu will be repeated after each operation.

Let us enter the option 1 in the submenu,

```
Enter your option: 1

Enter an element: Jan
A = {Mar, Jun, Sep, Jan}
..............................

Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Various displays
        9: Quit

Enter your option: 1

Enter an element: Jan
A = {Mar, Jun, Sep, Jan}
..............................
```

You can see in the above interactions, adding an element which is already inside the set, the final set remains unchanged; otherwise this distinct element is added to the set.

Let us explore option = 2

```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Various displays
        9: Quit

Enter your option: 2

Enter an element: Jun
Element Jun is in set
```

For option 3, the system just simply displays the cardinal number:

5

```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Various displays
        9: Quit

Enter your option: 3

No of elements in set is 4
.....................................
```

For option = 4:

```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Various displays
        9: Quit

Enter your option: 4

Equivalent sets info
        Set 1: {3, 6, 9, 1}
        set 2: {March, June, September, January}
.......................................................
```

You can continue to stay in the submenu or enter 9 to go back to the main menu.

Let us choose 9 to go back to the main screen to test other operations.

Let us now explore each of the options in the main menu:

In the main menu, you choose option 1:

```
Welcome to SIM Set Theory lesson

0: Properties of set
1. Union example
2. Intersection example
3. Subset example
4. Difference example
5. Complement example
6. Sets equality example
9. Quit

Your option: 1

Given sets
        A = {Jan, Jun, Oct, Nov, Feb, Jul}
        B = {Jun, Jul, Apr}
        Union of A and B = {Jan, Jun, Oct, Nov, Feb, Jul, Apr}
-------------------------------------------------
```

In option 1, the system randomly generates two sets and displays the union of these two sets.

The same is done for option 2, but evaluate the intersection of the two sets:

```
Your option: 2

Given sets
        A = {Jan, Aug, May, Oct, Sep, Dec, Jun}
        B = {Nov, Apr, Jul}
        Intersection of A and B = {}
-------------------------------------------------
```

In the above screen shot, you see the notation of an empty set. Let us see another screen shot for some non-empty intersections:

```
Your option: 2

Given sets
        A = {Dec, Nov, Jan, Oct, Jun, Jul, Aug, Mar}
        B = {Aug, Dec, Oct, Nov, Jun}
        Intersection of A and B = {Dec, Nov, Oct, Jun, Aug}
-------------------------------------------------
```

For option 3, the subset operation:

```
Your option: 3

Given sets
        A = {Jan, May, Mar, Jul, Feb, Apr, Jun, Oct, Dec}
        B = {Jun, Jul}

Conclusion
        A subset of B: false
        B subset of A: true
------------------------------------------------
```

For option 4, the difference of two sets;

```
Your option: 4

Given sets
        A = {Jun, Feb, Dec, Mar, Nov, Sep, Jul, Apr, May}
        B = {Sep, Nov, Jan, Aug, Oct, Jun}

        A - B = {Feb, Dec, Mar, Jul, Apr, May}
------------------------------------------------
```

Now, option 5, the complement of a set is done with the universal set. Our universal set is the set of calendar months. The following shows some of the interactions and displays:

```
Your option: 5

Given set
        A = {May, Nov, Jan, Jul, Jun, Oct, Mar, Feb}

        A' = {Apr, Aug, Sep, Dec}
------------------------------------------------
```

Option 6 is the set equality. The following shows some of the possible interactions and display:

```
Your option: 6

Given sets
        A = {May, Mar, Apr, Feb, Jan}
        B = {Feb, Mar, May, Apr, Jan}

Analysis
        A subset of B: true
        B subset of A: true

Conclusion
        A equals to B: true
------------------------------------------------
```

8

IMPORTANT TO NOTE

IMPORTANT

Put all your classes in a file called `YourName_A1.java` and make sure that this file can be compiled and can be executed. Upload **ONLY** this file to Moodle**. ALL ZIP FILE SUBMISSION WILL BE REJECTED**

No re-submission will be allowed after grading.

In the above file, remember to put down your name and also the following declaration (some similar contents):

**// Tell me if it is your own work, and whether you have passed your
// program to your friends etc etc etc
// and willing to accept whatever penalty given to you.**

- **Wrong file name -0.5 mark**
- **No declaration, no name etc -0.5 mark**
- **Failing to demo -1 mark**
- **Programs indentations and alignment of statements -0.5 mark**
- **Late penalty: -0.1 mark per hour**