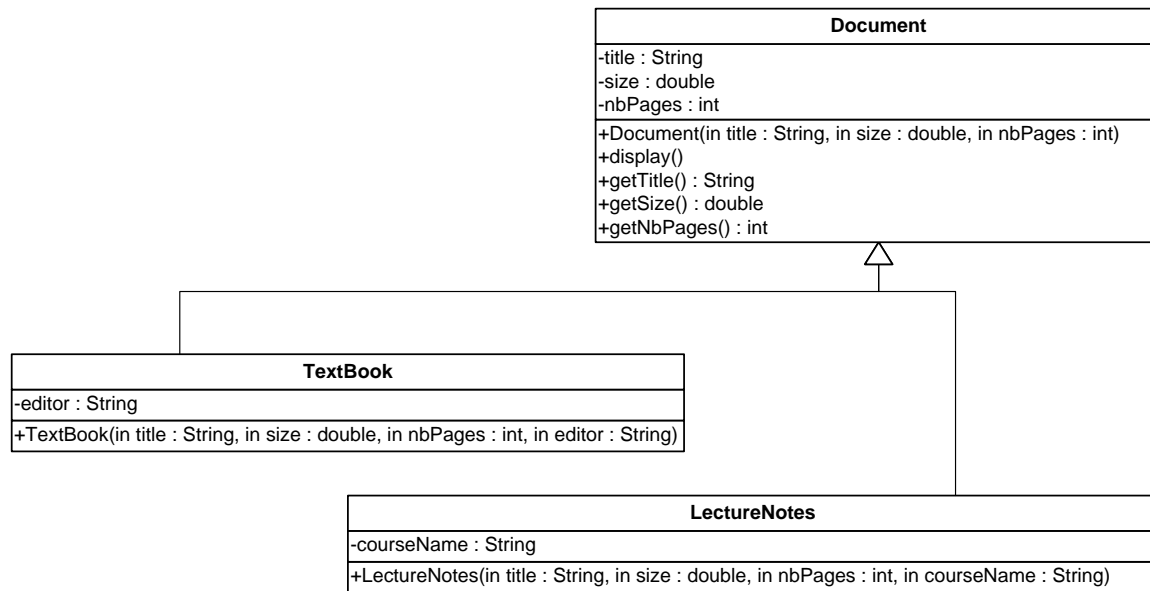


King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Midterm 1 Exam – Spring 2018

Exercise1:



Class *Document* :

- Attributes:
 - ***title***: the title of the document.
 - ***size***: the size in kilo-bytes of the document.
 - ***nbPages***: the number of pages of the document.
- Methods:
 - ***Document(title: String, size: double, nbPages: int)***: constructor.
 - ***display()***: this method displays all the attributes of the Document (TextBook or LectureNotes).

Class *TextBook* :

- Attributes:
 - ***editor***: the editor name of the textbook.
- Methods:
 - ***TextBook(title: String, size: double, nbPages: int, editor: String)***: constructor.

QUESTION: Translate into Java code the class ***Document*** and the class ***TextBook***.

Answer:

```
public class Document { ...../5
    private String title; //the title of the document.
    private double size; // the size in kilo-bytes of the document.
    private int nbPages;

    public Document(String t, double s, int nbP) { .....1
        title = t;
        size = s;
        nbPages = nbP;
    }

    public void display() { .....1
        System.out.println(title + size + nbPages);
    }

    public String getTitle() { .....1
        return title;
    }

    public double getSize() { .....1
        return size;
    }

    public int getNbPages() { .....1
        return nbPages;
    }
}
```

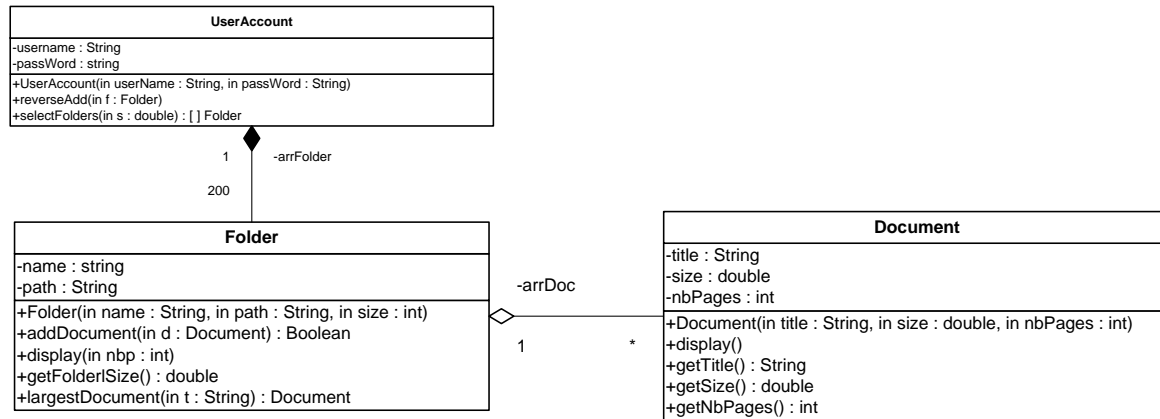
```
public class TextBook extends Document { .....1 ...../5
    private String editor;

    public TextBook(String t, double s, int nbP, String e) { ...../2
        super(t, s, nbP); .....1
        editor = e; .....1
    }

    public void display() { ...../2
        super.display(); .....1
        System.out.println(editor); .....1
    }
}
```

Exercise 2:

Let's consider the same class *Document* described in exercise 1.



Class *Folder*:

- Attributes:
 - **name**: the name of the folder.
 - **path**: the path of the folder.
- Methods:
 - **Folder (name: String, path: String, size: int)**: constructor. The parameter *size* defines the maximum number of documents in the folder.
 - **addDocument (d: Document)**: this method adds the document *d* to the folder. It returns true if the document is added successfully. Otherwise, it returns false.
 - **display(nbp: int)**: this method displays all documents of the folder having the number of pages greater than *nbp*.
 - **getFolderSize()**: this method returns the total size of all documents in the folder.
 - **largestDocument(t: String)**: this method returns the document with title *t* and having the largest number of pages.

QUESTION: Translate into Java code the class *Folder*.

Answer:

```
public class Folder { ...../25

    private String name; // the name of the folder.
    private String path; // the path of the folder.

    private Document arrDoc[]; .....1
    private int nbDocs; .....1

    public Folder(String n, String p, int size) { ...../2
        name = n;
        path = p;

        arrDoc = new Document[size]; .....1
        nbDocs = 0; .....1
    }

    public Folder(Folder f) { ...../4
        this.name = f.name;
        this.path = f.path;

        this.arrDoc = new Document[f.arrDoc.length]; .....1
        for (int i = 0; i < f.nbDocs; i++) .....1
            this.arrDoc[i] = f.arrDoc[i]; .....1

        this.nbDocs = f.nbDocs; .....1
    }

    public boolean addDocument (Document d) { ...../4

        if (nbDocs < arrDoc.length) { .....1
            arrDoc[nbDocs] = d; .....1
            nbDocs ++; .....1

            return true; .....0.5
        }
        else
            return false; .....0.5
    }

    public void display(int nbp) { ...../3
        for (int i = 0; i < nbDocs; i++) { .....1
            if (arrDoc[i].getNbPages() > nbp) .....1
                arrDoc[i].display(); .....1
        }
    }
}
```

```

public double getFolderSize() { ...../4
    double totals = 0.0; .....1

    for (int i = 0; i < nbDocs; i++) { .....1
        totals += arrDoc[i].getSize(); .....1
    }
    return totals; .....1
}

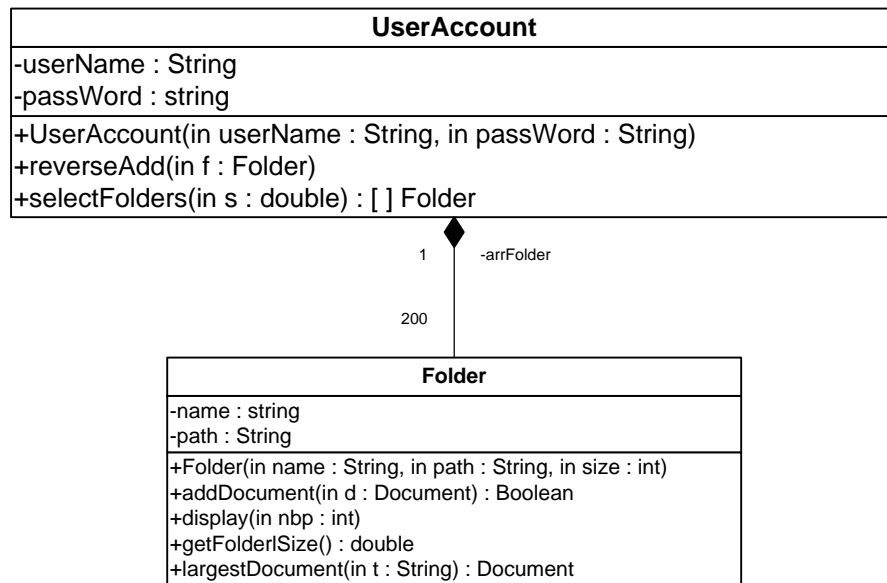
public Document largestDocument( String t) { ...../6
    Document largest = null; .....1

    for (int i = 0; i < nbDocs; i++) { .....1
        if (arrDoc[i].getTitle().equals(t)) { .....1
            if ( largest == null || .....1
                arrDoc[i].getNbPages()>largest.getNbPages())....1
                largest = arrDoc[i]; .....1
            }
        }
    }
    return largest;
}
}

```

Exercise 3:

Let's consider the same class **Folder** described in exercise 2.



class **UserAccount**:

○ Attributes:

- **userName**: the name of the user.
- **passWord**: the password of the user.

○ Methods:

- **UserAccount(userName: String, passWord: String)**: constructor.
- **reverseAdd(f: Folder)**: as shown in the picture below, this method inserts the folder *f* into array in the reverse order.

| | | | | | | | | | |
|---|---|---|---|---|---|---|----------|----------|----------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | | | Folder 3 | Folder 2 | Folder 1 |

- **selectFolders(s: double)**: this method returns an array containing all folders having size greater than *s*.

QUESTION: Translate into Java code the class **UserAccount**.

Answer:

```
public class UserAccount { ...../15
    private String userName; // the name of the user.
    private String passWord; // the password of the user.

    private Folder arrFold[]; .....1
    private int nbFolders; .....1

    public UserAccount(String u, String p) { ...../2
        userName = u;
        passWord = p;

        arrFold = new Folder[200]; .....1
        nbFolders = 0; .....1
    }

    public void reverseAdd(Folder f) { ...../4
        if (nbFolders < arrFold.length) { .....1
            arrFold[arrFold.length- 1- nbFolders] = new Folder(f); .1+1
            nbFolders ++; .....1
        }
    }

    public Folder[] selectFolders( double s) { ...../7
        Folder[] res = new Folder[nbFolders]; .....1
        int j = 0; .....1

        for (int i = 0; i < nbFolders; i++) { .....1
            if (arrFold[i].getFolderSize() > s) { .....1
                res[j] = arrFold[i]; .....1
                j++; .....1
            }
        }
        return res; .....1
    }
}
```