Lab06

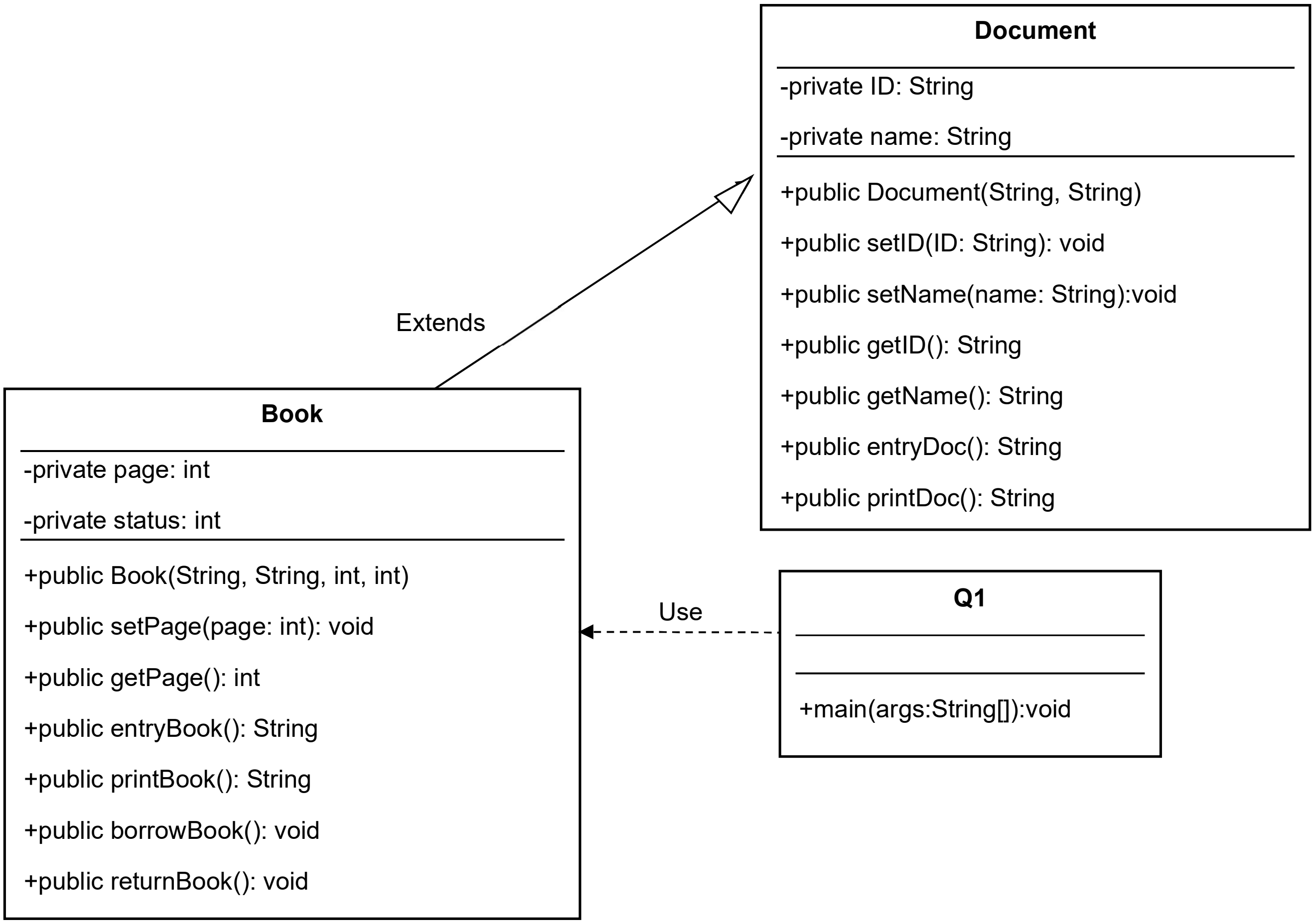
Tên project: Q1 và tạo thư mục nộp bài theo yêu cầu

```
solution.zip        // 5. Nén thư mục run & src, đặt tên solution.zip
    ├── run              // 4. Tạo thư mục run chứa file .jar
    │   └── Q1.jar       // File jar đã đóng gói (Clean and Build)
    └── src              // 3. Tạo thư mục src chứa Q1.zip
        └── Q1.zip       // 2. Nén dự án Q1 tạo ra từ NetBeans
            ├── build    // 1. Các file trong dự án NetBeans
            ├── dist
            ├── nbproject
            ├── src
            ├── test
            ├── build.xml
            └── manifest.mf
```

Nén thư mục nộp bài và submit lên EdutNext (Labs/Lab06)

Deadline:

Create 3 classes as below:



Document

-private ID: String

-private name: String

+public Document(String, String)

+public setID(ID: String): void

+public setName(name: String):void

+public getID(): String

+public getName(): String

+public entryDoc(): String

+public printDoc(): String

Book

-private page: int

-private status: int

+public Book(String, String, int, int)

+public setPage(page: int): void

+public getPage(): int

+public entryBook(): String

+public printBook(): String

+public borrowBook(): void

+public returnBook(): void

Q1

+main(args:String[]):void

1. Class **Document**
   - Fields
   - Contructor: set ID = “”, name = “”

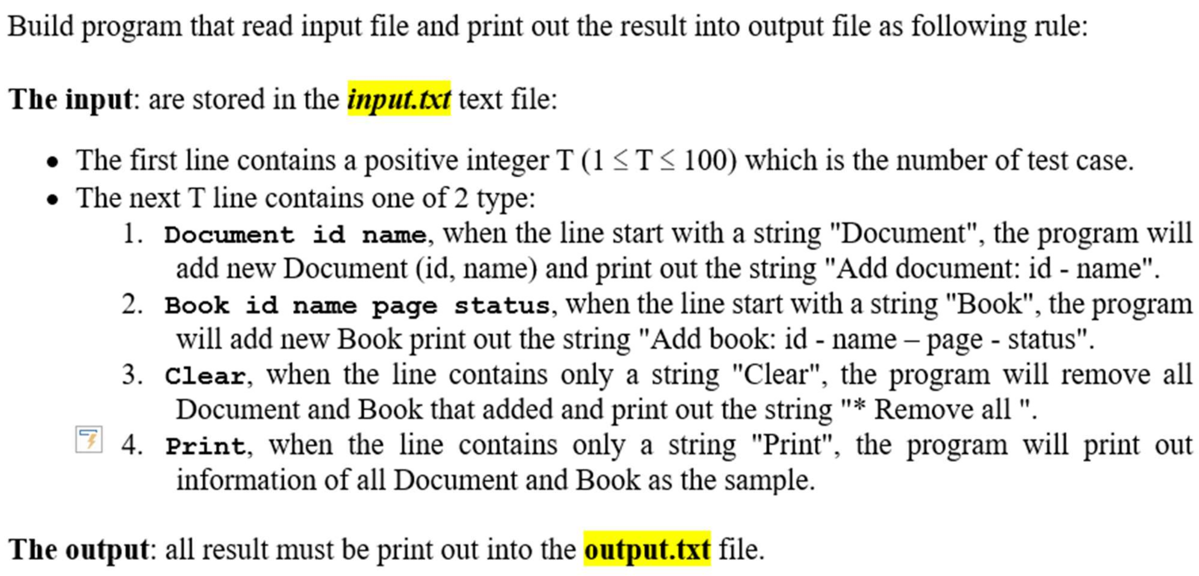
- get…(): return value of field
- set…(...): set value of field = …
- entryDoc(): return a string contains all field values
- printDoc(): return a string containing all field values

2. Class **Book**

- extends Document
- Fields
- Contructor: call super contructor, set page = 0 and status = 0
- get…(): return value of field
- set…(…): set value of field = …
- entryBook(): return a string contain all field values
- printBook(): return a string contains all field values

  if status = 0 then show “available” else show “unavailable”
- borrowBook(): set status = 1 (unavailable)
- returnBook(): set status = 0 (available)

3. Class **Q1**



Build program that read input file and print out the result into output file as following rule:

The input: are stored in the ***input.txt*** text file:

- The first line contains a positive integer T (1 ≤ T ≤ 100) which is the number of test case.
- The next T line contains one of 2 type:
  1. **`Document id name`**, when the line start with a string "Document", the program will add new Document (id, name) and print out the string "Add document: id - name".
  2. **`Book id name page status`**, when the line start with a string "Book", the program will add new Book print out the string "Add book: id - name – page - status".
  3. **`Clear`**, when the line contains only a string "Clear", the program will remove all Document and Book that added and print out the string "* Remove all ".
  4. **`Print`**, when the line contains only a string "Print", the program will print out information of all Document and Book as the sample.

The output: all result must be print out into the **output.txt** file.

Sample Input 1

input.txt

```
4
Document 001 CSoft
Document 002 SQLSoft
Print
Clear
```

output.txt

```
Add document: 001 – CSoft
Add document: 002 – SQLSoft
---Print---
001 – CSoft
002 – SQLSoft
----------
* Remove all
```

Sample Input 2

input.txt

```
8
Document 001 CSoft
Print
Book 002 JavaBook 200 0
Print
Clear
Document 001 SQLSoft
Book 002 CSSBook 200 1
Print
```

output.txt

```
Add document: 001 – Csoft
---Print---
001 – CSoft
Add book: 002 – JavaBook – 200 pages – available
---Print---
001 – CSoft
002 – JavaBook – 200 pages – available
----------
* Remove all
Add document: 001 – SQLSoft
Add book: 002 – CSSBook – 200 pages – unavailable
---Print---
001 – SQLSoft
002 – CSSBook – 200 pages – unavailable
```