

Môn học: Lập trình hướng đối tượng với Java

Bài thực hành số 10: Ngoại lệ và xử lý ngoại lệ

Câu 1. Xây dựng lớp `MyException` gồm các thuộc tính và phương thức như hình ở trang bên, biết rằng lớp `MyException` có:

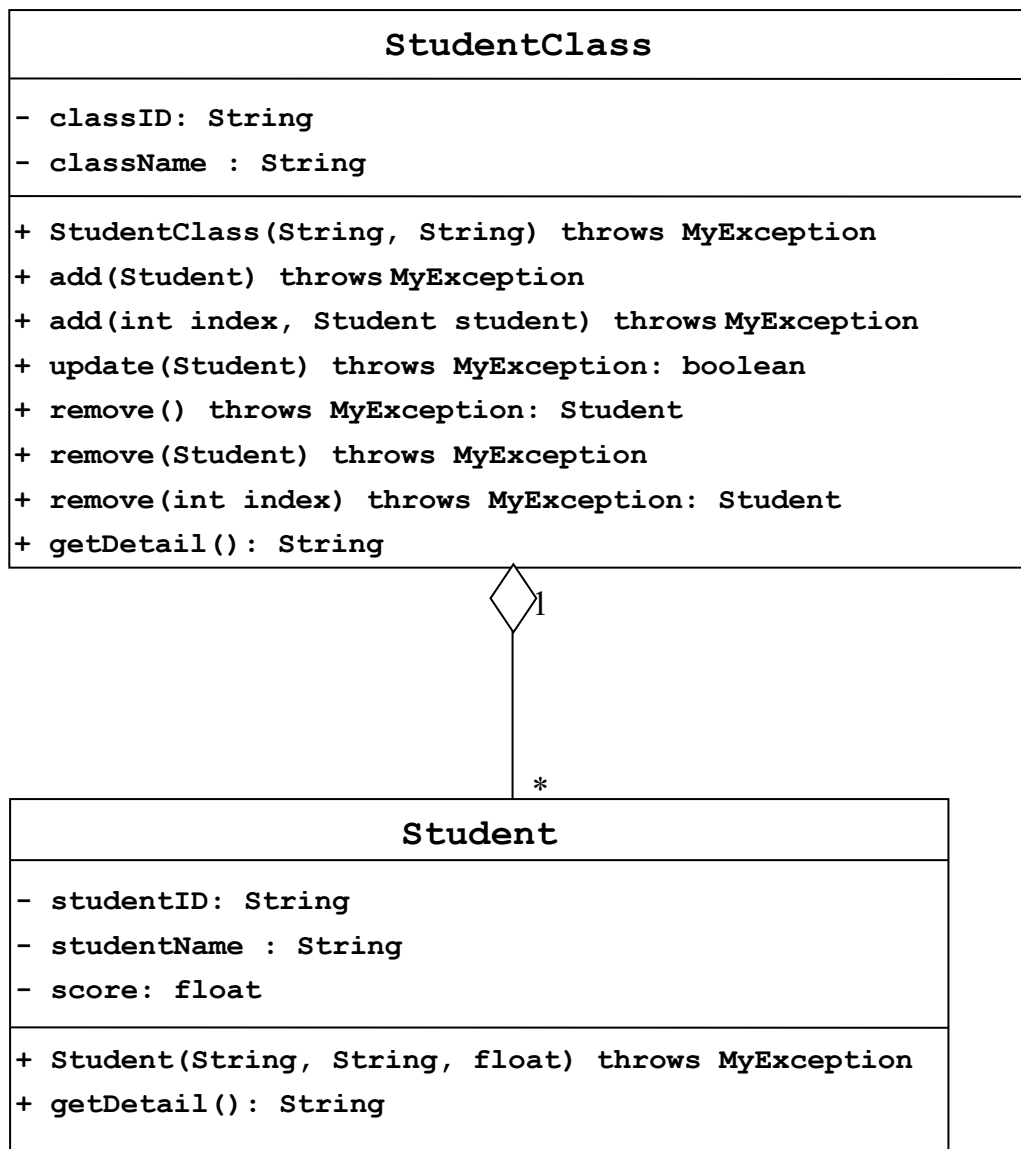
- a. 3 thuộc tính: `reason` chỉ lý do xảy ra ngoại lệ, `context` chỉ ngữ cảnh xảy ra ngoại lệ (xảy ra ở đâu, thông tin về nơi xảy ra ngoại lệ) và `info` là thông tin thêm cho ngoại lệ.
- b. 4 hằng tĩnh có giá trị như trong hình vẽ.
- c. Viết các phương thức `set` và `get` cho các thuộc tính `reason`, `context` và `info`; chú ý phương thức `setReason(String reasonType)` tùy thuộc vào tham số `reasonType` truyền vào mà thiết lập cho thuộc tính `reason` các giá trị tương ứng như sau:
 - i. Nếu `reasonType` có giá trị là `null` thì `reason` được thiết lập là “Null pointer exception”.
 - ii. Nếu `reasonType` có giá trị là hằng `EXIST` thì `reason` được thiết lập là “The id of the item you’ve entered is already existed!”
 - iii. Nếu `reasonType` có giá trị là hằng `NOT_EXIST` thì `reason` được thiết lập là “The item you want to update is not existed!”.
 - iv. Nếu `reasonType` có giá trị là hằng `INVALID` thì `reason` được thiết lập là “The item you’ve entered is not valid!”.
 - v. Nếu `reasonType` có giá trị là hằng `NO_ITEM` thì `reason` được thiết lập là “There is no item!”.
 - vi. Nếu `reasonType` có giá trị là hằng `NOT_EMPTY` thì `reason` được thiết lập là “The id of the item you’ve entered can not be empty!”.
- d. Phương thức `getMessage()` trả về tất cả các thông tin của ngoại lệ đó, giá trị trả về có dạng: “Reason: <giá trị của thuộc tính `reason`>; \nContext: <giá trị của thuộc tính>; \nAdditional Info: <giá trị của thuộc tính `info`>”. Chú ý xét các trường các thuộc tính `null` hoặc có giá trị “” thì không cần đưa vào giá trị trả về.
- e. Phương thức `toString()` trả về một chuỗi gồm: Tên lớp của đối tượng hiện tại, dấu “:” và giá trị của `getMessage()`

MyException
<pre> - reason: String - context : String - info: String + <u>EXIST = "0001"</u> + <u>NOT EXISTED = "0002"</u> + <u>INVALID = "0003"</u> + <u>NO ITEM = "0004"</u> + <u>NOT EMPTY = "0005"</u> </pre>
<pre> + MyException(String reason) + MyException(String reason, String context) + MyException(String reason, String context, String info) + MyException(Exception cause, String context) + getMessage(): String + toString(): String </pre>

- f. Viết các phương thức khởi tạo như trong hình vẽ, chú ý:
- Các phương thức khởi tạo sẽ gọi phương thức khởi tạo của lớp cha truyền các thông tin tương ứng bao gồm message (giống như kết quả của phương thức getMessage()) và cause (nếu có).
 - Phương thức MyException(Exception cause, String context) thiết lập cho các thuộc tính reason và info cho đối tượng ngoại lệ hiện tại từ đối tượng cause (nếu cause là một đối tượng của lớp MyException).

Câu 2. Viết mã nguồn cho lớp Student và StudentClass như hình vẽ biết rằng:

- Viết các phương thức set, get cần thiết cho 2 lớp Student và StudentClass.
- Các phương thức có throws ra ngoại lệ MyException thì cần có xử lý trong mã nguồn tương ứng với các reason đã liệt kê trong Bài 1 (nếu có). Chú ý:
 - Kiểm tra tính null của các đối tượng truyền vào để throw ra những ngoại lệ có thông tin tương ứng.
 - Mã của sinh viên trong tập hợp phải là duy nhất, ko cho phép có 2 sinh viên có cùng mã số.
 - score của sinh viên phải có giá trị nằm trong đoạn [0,4]. Nếu giá trị này được cập nhật không thỏa mãn thì throw ra ngoại lệ với thông tin tương ứng.



- c. Mối quan hệ giữa Student và StudentClass được cài đặt bằng một tập hợp, không được dùng mảng.
- d. Phương thức add(Student) thêm sinh viên vào cuối tập hợp. Nếu sinh viên thêm vào có mã số giống với sinh viên đã có trong tập hợp, throw ra ngoại lệ với các thông tin tương ứng.
- e. Phương thức add(int index, Student student) thêm student vào vị trí có chỉ số là index. Nếu sinh viên thêm vào có mã số giống với sinh viên đã có trong tập hợp, throw ra ngoại lệ với các thông tin tương ứng.
- f. Phương thức update(Student student) tìm kiếm student (theo mã) có trong tập hợp, nếu thấy thì cập nhật các thông tin còn lại vào sinh viên vừa tìm thấy, nếu không thấy thì throw ra ngoại lệ với các thông tin tương ứng.
- g. Phương thức remove() xóa sinh viên cuối cùng trong tập hợp (mới thêm vào nhất) và trả về sinh viên vừa xóa.

- h. Phương thức `remove(Student student)` tìm kiếm student trong tập hợp và xóa sinh viên đó khỏi tập hợp. Nếu không tìm thấy thì throw ra ngoại lệ với thông tin tương ứng.
- i. Phương thức `remove(int index)` xóa sinh viên có chỉ số là index khỏi tập hợp, trả về sinh viên vừa xóa. Nếu không tìm thấy sinh viên có chỉ số đưa vào thì throw ra ngoại lệ với thông tin tương ứng.
- j. Phương thức `getDetail()` của lớp `Student` trả về thông tin chi tiết về sinh viên đó. Phương thức `getDetail()` của lớp `StudentClass` trả về thông tin chi tiết của lớp cùng các thông tin chi tiết của sinh viên trong lớp đó.
- k. Viết lớp `Test` trong đó tạo các đối tượng của lớp `Student` và `StudentClass`; sử dụng các đối tượng đó trong việc gọi các phương thức đã viết trong các phần trên để minh họa cho việc bắt và xử lý ngoại lệ.

Câu 3. Tìm hiểu về `assertion` trong Java? Có mấy dạng `assertion`? Lấy ví dụ minh họa và giải thích cụ thể.