

JAVA – Assignment 7

Cách nộp bài: xem folder "Hướng dẫn nộp bài qua Git"

Chú ý:

Tạo 1 Project đặt tên là "TestingSystem_Assignment_7"

Tạo 3 package entity, backend, front-end

Trong phần backend sẽ tạo các class Exercise1, Exercise2, ... mỗi method là 1 Question

Trong front-end tạo các class Program1, Program2, Program3, ... để demo kết quả của các Exercise1, Exercise2, Exercise3, ...

Exercise 1: Static

Question 1: static variable

Khai báo 1 class student có các thuộc tính id, name, college
Với college là static variable.

Hãy khởi tạo các student sau:

Student có id = 1, name = "Nguyễn Văn A"

Student có id = 2, name = "Nguyễn Văn B"

Student có id = 3, name = "Nguyễn Văn C"

Và tất cả các student này đều học ở "Đại học bách khoa".

Dùng vòng for để in ra thông tin các student

Sau đó hãy chuyển các student này sang "Đại học công nghệ"

Dùng vòng for để in ra thông tin các student

Question 2: tiếp tục question 1

Bổ sung thuộc tính moneyGroup cho Student (moneyGroup là tiền quỹ lớp - dùng chung cho tất cả các student).

Hãy viết chương trình main() để mô tả các bước sau:

B1: Các Student sẽ nộp quỹ, mỗi Student 100k

B2: Student thứ 1 lấy 50k đi mua bim bim, kẹo về liên hoan

B3: Student thứ 2 lấy 20k đi mua bánh mì

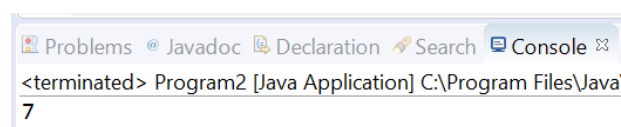
B4: Student thứ 3 lấy 150k đi mua đồ dùng học tập cho nhóm

B5: cả nhóm mỗi người lại đóng quỹ mỗi người 50k

In ra số tiền còn của nhóm tại mỗi bước

Question 3: static method

```
4 public static void main(String[] args) {  
5     int result = Math.max(5, 7);  
6     System.out.println(result);  
7 }  
8 }
```



- a) Viết class MyMath để thay thế cho class Math của java.
- b) Viết thêm method min(), sum vào class MyMath

Question 4: tiếp tục Question 1

Trong class Student

- a) Viết method cho phép thay đổi college
- b) Viết method cho phép lấy giá trị của college

Question 5:

Hãy viết chương trình đếm số Student được sinh ra (tham khảo code trên google)

Question 6: tiếp tục Question 5

Tạo class PrimaryStudent, SecondaryStudent, hãy viết chương trình đếm số lượng PrimaryStudent được sinh ra, SecondaryStudent được sinh ra.

Viết chương trình demo.

Khởi tạo 6 Student, trong đó có 2 PrimaryStudent và 4 SecondaryStudent, sau đó in ra số lượng Student, PrimaryStudent, SecondaryStudent được sinh ra.

Question 7: tiếp tục Question 6

Chỉ cho phép user tạo được tối đa 7 học sinh

Question 8:

Tạo class Configs có các static property
SO_LUONG_HINH_TOI_DA = 5;

Tạo class HìnhHoc, và class HìnhTron, HìnhChuNhat sẽ kế thừa từ class HìnhHoc, implement method tính chu vi, diện tích

Viết chương trình chỉ cho phép khởi tạo được tối đa 5 hình, nếu người dùng khởi tạo tới hình thứ 6 thì sẽ throw ra 1 custom Exception có tên là HìnhHocException có message là: "Số lượng hình tối đa là: " + Configs.SO_LUONG_HINH_TOI_DA

Question 9:

Thay đổi các method ở class ScannerUtils ở Assignment 6 thành static để tiện dùng hơn

Exercise 2 (Optional): Final

Question 1: final variable

Tạo class MyMath, khai báo final variable số PI = 3.14

Viết method sum(int a) và trả về tổng của a và PI

Thử thay đổi số PI = 3.15 trong method xem có được ko?

Question 2:

Tạo class Student có property id, name hãy thiết kế class Student sao cho khi đã khởi tạo Student thì id là không thể thay đổi trong suốt quá trình chạy chương trình

Question 3: Tiếp tục Question 2 (final method)

Student bắt buộc phải học bài, và việc học bài là như nhau đối với các học sinh → ta viết method void study(), bên trong method ta sẽ in ra text "Đang học bài..."

Tiếp theo ta có các loại Student như sau: PrimaryStudent, SecondaryStudent, và có thể còn có nhiều loại học sinh khác trong tương lai, tất cả các student này đều kế thừa Student

Hãy thiết kế chương trình để tất cả các học sinh đều có method study() như nhau (nghĩa là PrimaryStudent, SecondaryStudent không thể override được method study())

Question 4: Tiếp tục Question 3 (final class)

Hãy thiết kế chương trình sao cho không có class nào có thể kế thừa từ PrimaryStudent, SecondaryStudent

Exercise 3 (Optional): File

Tạo class FileManager ở package utils, sau đó tất cả các Question trong Exercise này sẽ viết vào trong class FileManager, với mỗi Question sẽ viết demo ở front-end

Chú ý: tất cả các error message nên để là constant (static final String)

Question 1: Check File is exists

Tạo 1 method có đầu vào là String path để check xem path đó có tồn tại hay không.

VD: path = "C:\Users\pc\Desktop\Test.txt"

Gợi ý: Tạo method **boolean isFileExists(String pathFile)**

Question 2: Create new file

Tạo 1 method để có thể tạo được file.

Chú ý:

Kiểm tra xem file đã tồn tại hay chưa, nếu file đã tồn tại thì sẽ throw ra Exception "Error! File Exist."

Gợi ý:

Tạo method **void createNewFile(String pathFile)**

Vào tạo thêm method **void createNewFile(String path, String fileName)**

Question 4: Delete file

Tạo 1 method để có thể delete file

Chú ý:

Kiểm tra xem file đã tồn tại hay chưa, nếu file chưa tồn tại thì sẽ throw ra Exception "Error! File Not Exist."

Gợi ý: Tạo method **void deleteFile(String pathFile)**

Question 5: Check path is File or Folder

Tạo 1 method để kiểm tra xem path có thể là File hay Folder.

Gợi ý: Tạo method **boolean isFolder(String path)**

Question 6: Get all File name of Folder

Tạo 1 method để lấy ra tất cả các tên file trong 1 Folder.

Chú ý:

Kiểm tra xem path nhập vào có phải là folder hay không, nếu không phải thì sẽ throw ra Exception "Error! Path is not folder."

Gợi ý: Tạo method **List<String> getAllFileName(String path)**

Question 7: Copy File

Tạo 1 method để copy file.

Chú ý:

Nếu file không tồn tại thì sẽ throw ra Exception "Error! Source File Not Exist."

Nếu file đích đã tồn tại thì sẽ throw ra Exception "Error! newPath has File same name."

Gợi ý:

Tạo method **void copyFile(String sourceFile, String destinationPath, String newName)**

Và tạo method **void copyFile(String sourceFile, String newPath)**. Với File mới sẽ có tên cùng với file cũ

Question 8: Moving file

Tạo 1 method để di chuyển file sang folder khác.

Chú ý:

Nếu file không tồn tại thì sẽ throw ra Exception "Error! File Not Exist."

Phải xóa file cũ đi

Gợi ý:

Tạo method **void moveFile(String sourceFile, String destinationPath)**

Question 9: Rename File

Tạo 1 method để đổi tên file.

Chú ý:

Nếu file không tồn tại thì sẽ throw ra Exception "Error! File Not Exist."

Nếu newName đã tồn tại thì sẽ throw ra Exception "Error! Name is Exist."

Gợi ý:

Tạo method **void renameFile(String pathFile, String newName)**

Question 10: Create new folder

Tạo 1 method để có thể tạo được folder.

Chú ý:

Kiểm tra xem folder đã tồn tại hay chưa, nếu folder đã tồn tại thì sẽ throw ra Exception "Error! Folder Exist."

Gợi ý: Tạo method **void createNewFolder(String newPathFolder)**

Question 11: Download File

Hãy viết 1 method để có thể download được file ở trên mạng

Chú ý:

Nếu folder không tồn tại thì sẽ tạo folder.

Nếu tên file download đã tồn tại trong folder thì sẽ thêm (1), (2) vào sau tên file (giống download file ở window).

Gợi ý:

Tạo method **void downloadFile(String fileLink, String folder)**

Academy

```

public boolean downloadFile(String link, String folderSave)
    throws MalformedURLException, IOException {
    // if folder not Exist or not folder
    File folder = new File(folderSave);
    if (!folder.exists() || !folder.isDirectory()) {
        // show message error
        System.out.println(FOLDER_NOT_EXIST_OR_NOT_FOLDER);
        return false;
    }
    // if folder exist
    // get name file
    String s[] = link.split("/");
    String name = s[s.length - 1];
    folderSave = folderSave + "/" + name;
    // create connection to URL
    URL url = new URL(link);
    // open connection
    URLConnection connection = url.openConnection();
    // get size of file
    int size = connection.getContentLength();
    // read file from Internet use InputStream
    InputStream in = connection.getInputStream();
    // save file use FileOutputStream
    FileOutputStream output = new FileOutputStream(folderSave);
    int byteDownloaded = 0;
    byte[] b = new byte[1024];
    // get length of file. If not read then length = -1
    int length = in.read(b);
    while (length != -1) {
        byteDownloaded += length;
        // print % byte downloaded
        System.out.println(byteDownloaded * 100f / size + "%");
        // write content downloaded from position 0 -> length to output
        output.write(b, 0, length);
        length = in.read(b);
    }
    // close
    output.close();
    in.close();
    // show message
    System.out.println(DOWNLOAD_FILE_SUCCESS);
    return true;
}

```

Code tham khảo

Exercise 4 (Optional): IO Stream

Tạo class IOManager ở package utils, sau đó tất cả các Question trong Exercise này sẽ viết vào trong class IOManager, với mỗi Question sẽ viết demo ở front-end

Chú ý: tất cả các error message nên để là constant (static final String)

Question 1: Tạo method read File

Tạo 1 method có đầu vào là String filePath và return ra nội dung trong file.

Chú ý:

Nếu file không tồn tại thì sẽ throw ra Exception "Error! File Not Exist." (sử dụng FileManager ở trên để check)

Gợi ý: Tạo method **String readFile(String pathFile)**

Question 2: Tạo method write content to File

Tạo 1 method để write file

Chú ý:

Nếu file không tồn tại thì sẽ throw ra Exception "Error! File Not Exist." (sử dụng FileManager ở exercise trên để check)

Gợi ý: Tạo method **void writeFile(String pathFile, isContinuing String content)**

Nếu isContinuing = false thì sẽ xóa hết nội dung trong file đi và write content mới vào

Nếu isContinuing = true thì sẽ ghi content vào cuối file

Question 3: Tạo method write Object

Hãy viết 1 method để có thể write object

Chú ý:

Nếu object = null thì sẽ throw ra Exception "Error! Object is Null."

Nếu folder không tồn tại thì sẽ tạo folder.

Nếu file đã tồn tại trong folder thì sẽ ghi đè lên file cũ

Gợi ý: Tạo method **void writeObject(Object object, String path, String fileName)**

Question 4: Tạo method read Object

Tạo 1 method có đầu vào là String filePath và return ra Object

Chú ý:

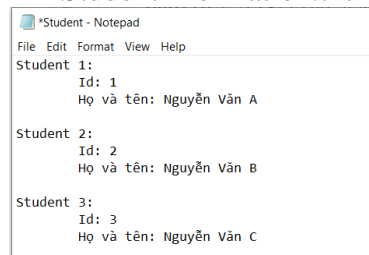
Nếu file không tồn tại thì sẽ throw ra Exception "Error! File Not Exist." (sử dụng FileManager ở trên để check)

Gợi ý: Tạo method **Object readObject(String filePath)**

Exercise 5: Demo File & IO Stream

Question 1: Tạo class Student có property id, name.

- Sau đó khởi 3 instance từ console (sử dụng ScannerUtils)
- Write 3 student này ra file tên là StudentInformation.txt
- Sau đó đọc thông tin file StudentInformation.txt và in ra màn hình



```
*Student - Notepad
File Edit Format View Help
Student 1:
Id: 1
Họ và tên: Nguyễn Văn A

Student 2:
Id: 2
Họ và tên: Nguyễn Văn B

Student 3:
Id: 3
Họ và tên: Nguyễn Văn C
```

Question 2: Tạo LogUtils (sử dụng write Object Exception)

Tạo 1 custom Exception Object, sau đó tại hàm main sẽ bắt exception ko mong muốn và lưu thông tin Exception.ser file

Các bước làm sẽ như sau:

B1: Tạo 1 object MyException sẽ extends Exception, trong MyException sẽ chứa các thông tin như: message, reason, StackTrace, time (thời gian bị Exception), ...

B2: Sau đó tạo class LogUtils có 1 vài method như sau:

1. void writeLog (String message, String reason, StackTrace, Datetime time) (gợi ý: sử dụng IOManager ở trên để write file)
2. void writeLog (MyException exception)

B3: đặt try catch ở method main() hoặc ở bất kỳ chỗ nào muốn lưu lại thông tin exception

B4: Tạo method để đọc thông tin trong file Exception và in ra màn hình

B5: Demo chương trình

Question 3 (Optional):

Thiết kế bài lô đề của exercise trước với những cải tiến về handling exception, đặt các property static, final cho hợp lý và thêm các chức năng sau đây:

- a) Đăng ký tài khoản (thông tin các tài khoản sẽ lưu vào file)
Có 2 loại tài khoản admin và user.
- b) Đăng nhập vào hệ thống
- c) Đăng xuất tài khoản
- d) Cho phép nạp tiền vào tài khoản (lưu thông tin vào file)
- e) Lưu thông tin kết quả lô đề từng ngày vào file
- f) Lưu thông tin người dùng chơi lô đề vào file
- g) Tạo thêm chức năng sau:
 - 1) Chỉ có admin mới có thể xem thông tin tất cả các tài khoản (giống table ở SQL khi SELECT * Account, ẩn mật khẩu đi)
 - 2) Người dùng có thể xem lại thông tin lô đề từng ngày (hệ thống có hỗ trợ chức năng tìm kiếm theo ngày, và lọc chỉ xem từ ngày nào tới ngày nào)

Chú ý:

- Viết đúng coding convention
- Tuân thủ các best practice
- Không chép bài người khác (sẽ có hình thức phạt nếu bị phát hiện)