



JAVA Assignments

Program Code	Fresher Academy
Version	1.0
Effective Date	

Hanoi, 07/2017

Assigment 1: Thiết kế một ứng dụng tính chu vi, diện tích.

Mục tiêu

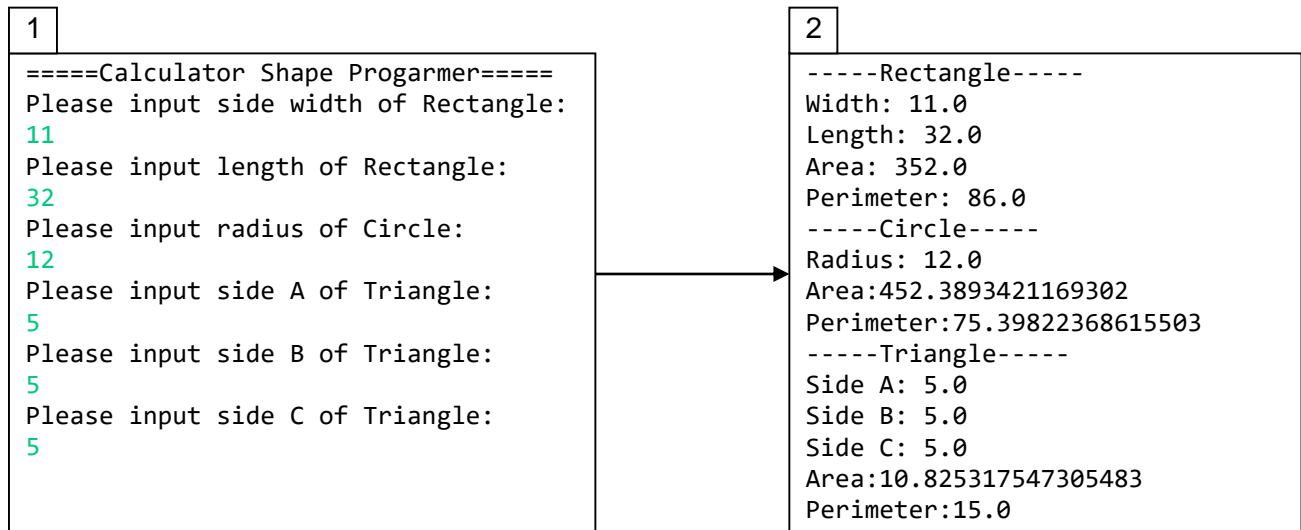
Học viên hiểu được thế nào là:

- Một Super Class thế nào là kế thừa.
- Thế nào là một Abstract Class.
- Thế nào là tính đóng gói.
- Thế nào là tính đa hình.

Mô tả bài toán

Thiết kế ứng dụng tính chu vi, diện tích hình Tròn, Chữ nhật, Tam giác.

Yêu cầu giao diện



Yêu cầu chức năng

Điều kiện tiên đề:

- Người dùng chạy ứng dụng, chương trình yêu cầu người dùng nhập vào thông tin các cạnh của hình Vuông, Tròn, Tam giác.
- In ra thông tin các cạnh và diện tích và chu vi của các hình Vuông, Tròn, Tam giác.

Chức năng 1: Tính chu vi

- Chương trình tính chu vi của hình.
- Tên hàm trong code: `public double getPerimeter()`
 - Giá trị trả về: chu vi của hình.

Chức năng 2: Tính diện tích.

- Chương trình tính diện tích của hình.
- Tên hàm trong code: `public double getArea()`
 - Giá trị trả về: Diện tích của hình.

Chức năng 3: in thông tin hình.

- Chương trình in ra thông tin các cạnh, chu vi và diện tích.
- Tên hàm trong code: `public void printResult()`
 - Giá trị trả về: void.

Hướng dẫn cho học viên

Gợi ý:

- Thiết kế 1 abstract class Shape chứa ba phương thức `printResult`, `getPerimeter` và `getArea`.
- Các class Tam giác, Chữ nhật, Tròn kế thừa Shape.
- Xây dựng các hình gồm các thuộc tính của hình tròn(radius), chữ nhật(width, length), tam giác(sideA, sideB, sideC) và tạo thêm các phương thức getter và setter.
- Override lại các phương thức của lớp Shape.
- Tính diện tích hình tròn bằng công thức Heron:

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

Sử dụng `Math.sqrt()`

- Tính diện tích hình tròn với `Pi = Math.PI`

Assigment 2: Thiết kế một ứng dụng phân tích đường dẫn file

Mục tiêu

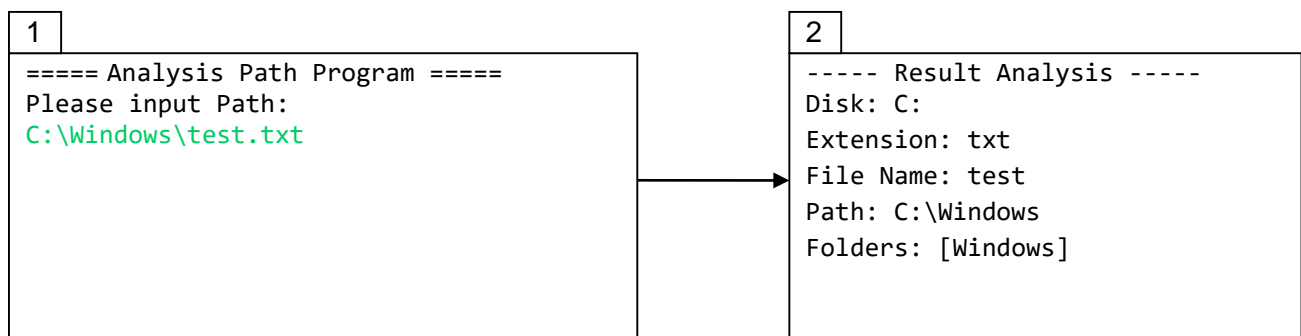
Học viên hiểu được:

- Class String.
- Cách sử dụng String manipulation.

Mô tả bài toán

Nhập vào đường dẫn của một file lấy ra tên ổ đĩa, tên file, loại file, các folder và đường dẫn file, in thông tin ra màn hình

Yêu cầu giao diện



Yêu cầu chức năng

Điều kiện tiên đề:

- Người dùng chạy ứng dụng, chương trình yêu cầu người dùng nhập vào thông tin đường dẫn file.
- In ra thông tin đường dẫn file, loại file, tên file.

Chức năng 1: Lấy đường dẫn file

- Tên hàm trong code: `public String getPath()`
 - Giá trị trả về: đường dẫn file.

Chức năng 2: Lấy tên file.

- Tên hàm trong code: `public String getFileName()`
 - Giá trị trả về: tên file.

Chức năng 3: Lấy loại file.

- Tên hàm trong code: `public String getExtension()`
 - Giá trị trả về: lấy về loại file.

Chức năng 4: Lấy về Tên ổ đĩa.

- Tên hàm trong code: `public String getDisk()`
 - Giá trị trả về: tên ổ đĩa.

Chức năng 5: Lấy các tên thư mục.

- Tên hàm trong code: `public String[] getFolders()`
 - Giá trị trả về: các tên Folder.

Hướng dẫn cho học viên

Gợi ý:

- Sử dụng các hàm `LastIndexOf`, `IndexOf`, `SubString`, `Split` của `String` để xử lý chuỗi và lấy ra kết quả.

Assigment 3: Tìm điểm trung bình lớn nhất, nhỏ nhất và xếp loại sinh viên.

Mục tiêu

Học viên nắm được các class và interface trong Collection Framework.

Mô tả bài toán

- Thiết kế bài toán nhận đầu vào:
 - Thông tin sinh viên bao gồm tên sinh viên tên lớp điểm Toán, Lý, Hóa.
- In ra màn hình những thông tin:
 - Xếp Loại sinh viên
 - $\text{Mark} < 4$: D
 - $4 \leq \text{mark} < 6$: C
 - $6 \leq \text{mark} \leq 7.5$: B
 - $\text{Mark} > 7.5$: A.
 - Thống kê loại sinh viên theo %.
 - Đưa ra điểm trung bình lớn nhất, nhỏ nhất.

Yêu cầu giao diện

1

```
===== Management Student Program =====
Name:Nghia
Classes:FU1
Maths:11
Maths is less than equal ten
Maths:-1
Maths is greater than equal zero
Maths:
Maths is digit
Maths:10
Chemistry:11
Chemistry is less than equal ten
Chemistry:-1
Chemistry is greater than equal zero
Chemistry:
Chemistry is digit
Chemistry:10
Physics:11
Physics is less than equal ten
Physics:-1
Physics is greater than equal zero
Physics:
Physics is digit
Physics:10
Do you want to enter more student information?(Y/N):Y
Name:Nghia 2
Classes:FU1
Maths:10
Chemistry:10
Physics:10
Do you want to enter more student information?(Y/N):N
```

2

```
----- Student1 Info -----
Name:Nghia
Classes:FU1
AVG:10.0
Type:A
----- Student2 Info -----
Name:Nghia 2
Classes:FU1
AVG:10.0
Type:A
-----Mark Info-----
Mark Min:10.0
Mark Max:10.0
-----Classification Info -----
A: 100.0%
B: 0.0%
C: 0.0%
D: 0.0%
```

Yêu cầu chức năng

Điều kiện tiên đề:

- Người sử dụng chạy ứng dụng, chương trình hiển thị màn hình yêu cầu người sử dụng nhập thông tin sinh viên.
- Tiếp đó, chương trình hỏi người dùng có muốn nhập sinh viên nữa không?
- Nếu có thì cho nhập tiếp, nếu không thì in ra kết quả.

Chức năng 1: Nhấp thông tin sinh viên

- Tên hàm trong code: `Student createStudent(String name, String classes, double maths, double chemistry, double physics)`
 - Đầu vào:
 - name: Tên sinh viên
 - classes: Lớp học.
 - maths: Điểm môn Toán.
 - chemistry: Điểm môn Hóa.
 - physics: Điểm môn Lý
 - Giá trị trả về:
 - Đối tượng sinh viên.

Chức năng 2: Xếp loại sinh viên.

- Tên hàm trong code: `List<Student> averageStudent(List<Student> students)`
 - Đầu vào:
 - students: Thông tin các sinh viên chưa được xếp loại
 - Giá trị trả về: các sinh viên đã được xếp loại.

Chức năng 3: Thống kê sinh viên.

- Tên hàm trong code: `HashMap<String, Double> getPercentTypeStudent(List<Student> students)`
 - Đầu vào:
 - students: thông tin sinh viên đã được xếp loại.
 - Giá trị trả về: Thông tin % các loại sinh viên theo key A, B, C, D.

Chức năng 4: Đưa ra average lớn nhất, nhỏ nhất.

- Tên hàm trong code: `HashMap<String, Double> getMaxAndMinAVG(List<Student> students)`
 - Đầu vào:
 - `students`: thông tin sinh viên đã được xếp loại.
 - Giá trị trả về: điểm trung bình key MAX, MIN

Hướng dẫn cho học viên

Gợi ý:

Tạo ra một lớp sinh viên bao gồm các thuộc tính:

- Tên, Lớp, điểm toán, lý, hóa, điểm trung bình, loại

Tạo ra một lớp tính điểm, xếp loại thống kê sinh viên rồi set kết quả vào đối tượng sinh viên

Tìm Max Min: gán giá trị `max=min=điểm trung bình`

Duyệt mảng :

Nếu điểm trung bình > max thì gán giá trị cho max.

Min làm ngược lại.

Đưa ra các thống kê:

A: ? %

B: ? %

C: ? %

D: ? %

Assignment 4: Nhập, sắp xếp và hiển thị thông tin Student.

Mục tiêu

Hiểu được cách sử dụng cú pháp for, if, try catch.

Hiểu được cách kiểm tra giá trị có phải là một số hay không.

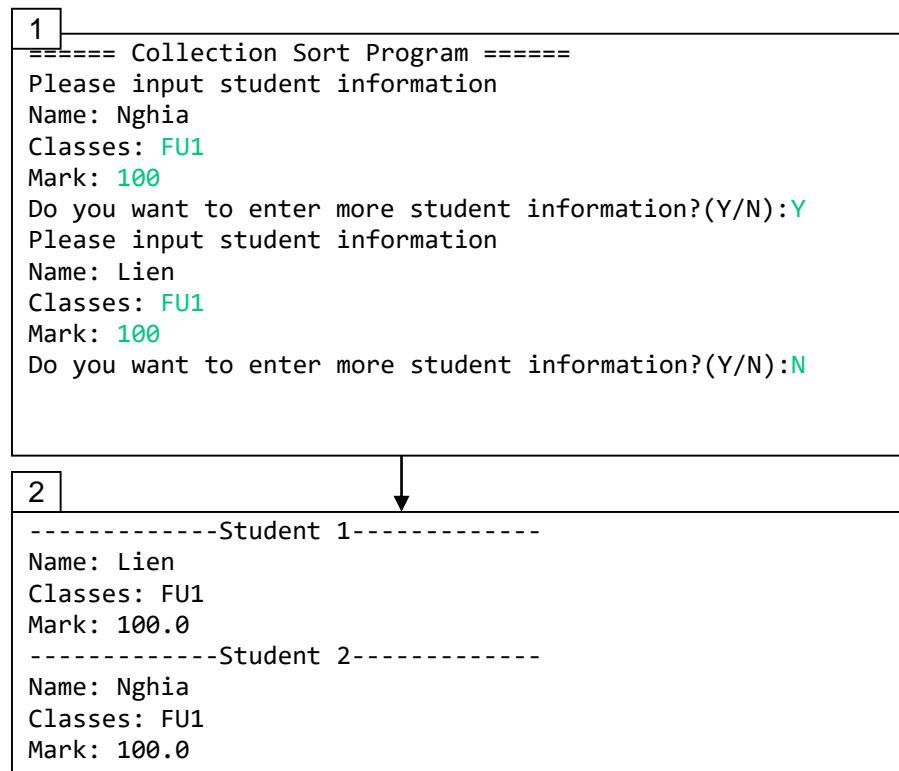
Hiểu được cách sử dụng Object.

Hiểu được cách sử dụng Collection

Mô tả bài toán

Viết một chương trình cho phép nhập thông tin Sinh viên(Name, Class, Mark) sau đó dùng collection để sắp xếp theo tên sinh viên từ A-Z, in thông tin sinh viên ra màn hình.

Yêu cầu giao diện



Yêu cầu chức năng

Điều kiện tiên đề:

- Người sử dụng chạy ứng dụng, chương trình hiển thị màn hình yêu cầu người sử dụng nhập vào thông tin Sinh viên bao gồm tên, lớp, điểm
- Tiếp đó, chương trình hỏi người dùng xem có muốn nhập thêm sinh viên nữa không.?
- Nếu có thì tiếp tục cho nhập thông tin sinh viên
- Nếu không thì thực hiện sắp xếp sinh viên và in thông tin lên màn hình, kết thúc chương trình.

Chức năng 1: Nhập thông tin của Student

- Chương trình yêu cầu nhập tên, địa chỉ, lương của Student.
- Tên hàm trong code: `List<Student> sortStudent(List<Student> students)`
 - Đầu vào:
 - `students`: danh sách sinh viên đã nhập
 - Giá trị trả về: danh sách sinh viên đã được sắp xếp

Chức năng 2: Hiển thị thông tin của Student.

- Chương trình thực hiện hiển thị tất cả thông tin của Student vừa nhập.
- Tên hàm trong code: `void display(List<Student> students)`
 - Đầu vào:
 - `students` : Danh sách sinh viên đã được sắp xếp.
 - Giá trị trả về: `void`.

Hướng dẫn cho học viên

Gợi ý:

Tạo class có tên là Student và có các thuộc tính sau đây:

`private String name;`

`private float mark;`

`private String classes;`

+ Tạo các constructor:

- Default constructor
- Constructor có tham số

+ Tạo các hàm get/set cho các thuộc tính của Student

Tạo class Student comparator implement Comparator ghi đè lại phương thức compare

+ Sử dụng Collection.sort() để sắp xếp các Student theo tên từ A-Z

Tạo class Main có hàm main() để test class trên và thực hiện như sau:

- Yêu cầu người dùng nhập vào thông tin sinh viên
- Nếu Điểm của sinh viên không phải là số thì bắt người dùng nhập lại
- Gọi hàm sortStudent() để sắp xếp danh sách sinh viên.
- Gọi hàm display() để hiển thị thông tin danh sách sinh viên.

Assignment 5: Nhập, sắp xếp và hiển thị thông tin Student.

Mục tiêu

Hiểu được cách sử dụng cú pháp for, if, try catch.

Hiểu được cách kiểm tra giá trị có phải là một số hay không.

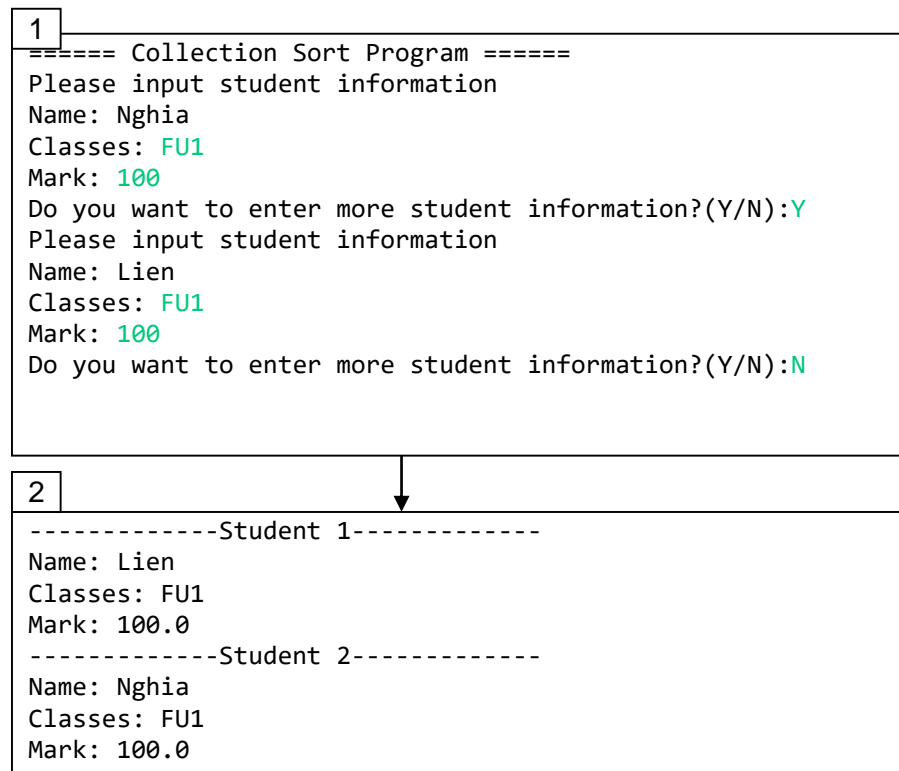
Hiểu được cách sử dụng Object.

Hiểu được cách sử dụng Collection

Mô tả bài toán

Viết một chương trình cho phép nhập thông tin Sinh viên(Name, Class, Mark) sau đó dùng collection để sắp xếp theo tên sinh viên từ A-Z, in thông tin sinh viên ra màn hình.

Yêu cầu giao diện



Yêu cầu chức năng

Điều kiện tiên đề:

- Người sử dụng chạy ứng dụng, chương trình hiển thị màn hình yêu cầu người sử dụng nhập vào thông tin Sinh viên bao gồm tên, lớp, điểm
- Tiếp đó, chương trình hỏi người dùng xem có muốn nhập thêm sinh viên nữa không.?
- Nếu có thì tiếp tục cho nhập thông tin sinh viên
- Nếu không thì thực hiện sắp xếp sinh viên và in thông tin lên màn hình, kết thúc chương trình.

Chức năng 1: Nhập thông tin của Student

- Chương trình yêu cầu nhập tên, địa chỉ, lương của Student.
- Tên hàm trong code: `List<Student> sortStudent(List<Student> students)`
 - Đầu vào:
 - `students`: danh sách sinh viên đã nhập
 - Giá trị trả về: danh sách sinh viên đã được sắp xếp

Chức năng 2: Hiển thị thông tin của Student.

- Chương trình thực hiện hiển thị tất cả thông tin của Student vừa nhập.
- Tên hàm trong code: `void display(List<Student> students)`
 - Đầu vào:
 - `students` : Danh sách sinh viên đã được sắp xếp.
 - Giá trị trả về: `void`.

Hướng dẫn cho học viên

Gợi ý:

Tạo class có tên là Student và có các thuộc tính sau đây:

`private String name;`

`private float mark;`

`private String classes;`

+ Tạo các constructor:

- Default constructor
- Constructor có tham số

+ Tạo các hàm get/set cho các thuộc tính của Student

Tạo class Student comparator implement Comparator ghi đè lại phương thức compare

+ Sử dụng Collection.sort() để sắp xếp các Student theo tên từ A-Z

Tạo class Main có hàm main() để test class trên và thực hiện như sau:

- Yêu cầu người dùng nhập vào thông tin sinh viên
- Nếu Điểm của sinh viên không phải là số thì bắt người dùng nhập lại
- Gọi hàm sortStudent() để sắp xếp danh sách sinh viên.
- Gọi hàm display() để hiển thị thông tin danh sách sinh viên.

--- THE END ---