
LẬP TRÌNH CƠ BẢN VỚI JAVA

Bài tập tuần 13

Bài 1. Sử dụng giao diện ***Shape*** trong chương trình ví dụ (tuần 11) hãy cài đặt lớp ***Triangle*** mô tả kiểu dữ liệu của các đối tượng tam giác. Lớp ***Triangle*** có ba thuộc tính là ba cạnh của tam giác.

Xây dựng một chương trình nhập một danh sách các đối tượng hình học gồm các hình vuông và hình tam giác. Sắp xếp lại danh sách tăng dần theo diện tích. Hình có diện tích nhỏ nhất là tam giác hay hình vuông?

Ví dụ:

Đầu vào:

Hình tam giác: 3 4 5

Hình tam giác: 4 5 4

Hình vuông: 3.5

Hình vuông: 2

Hình tam giác: 3 3 3

Đầu ra:

Danh sách các hình theo diện tích là:

[3, 3, 3]

[2]

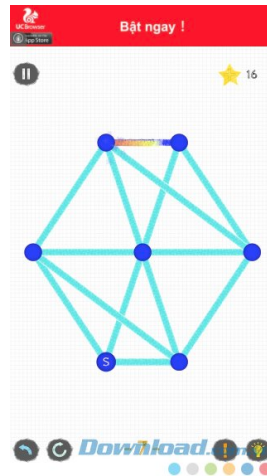
[3, 4, 5]

[4, 5, 4]

[3.5]

Hình có diện tích nhỏ nhất là hình tam giác

Bài 2. One Touch Drawing for iOS là trò chơi giải đố rất gần gũi với tuổi học trò với tên gọi "Vẽ hình 1 nét". Đây là một trò chơi trí tuệ tuy đơn giản nhưng lại cực kỳ gây nghiện. Luật chơi cũng khá đơn giản, dựa vào hình mẫu, người chơi One Touch Drawing cho iOS sẽ phải vẽ lại bằng một nét mà không được nhấc tay lên khỏi màn hình hay đi một đường 2 lần.



Để cai nghiệm, sinh viên Khoa Toán - Cơ - Tin học quyết định xây dựng một chương trình tự động tìm đáp án của trò chơi. Đó là chương trình có lớp đối tượng cho phép biểu diễn một đồ thị tương ứng với một hình mẫu, và tìm một đường đi Euler trên đồ thị nếu có.

Yêu cầu:

- Tạo lớp **Graph** là kiểu dữ liệu cho các đối tượng đồ thị (với thông tin về các đỉnh và các cạnh của đồ thị). Lớp này có chứa phương thức cho phép in ra một đường đi Euler nếu có.
- Viết lớp **GraphTest** để nhập thông tin cho một đối tượng đồ thị, sau đó in ra đường đi Euler của đồ thị vừa nhập.

Bài 3. Cho một dãy gồm có n phần tử khác nhau đôi một. Hãy đếm xem có bao nhiêu cách hoán vị của dãy đã cho.

Yêu cầu: Xây dựng lớp `Permutations1` có thuộc tính là một số nguyên dương n , và có phương thức `int count()` cho phép đếm số hoán vị.

Ví dụ:

Đầu vào:

3

Đầu ra:

Co 6 hoan vi

Bài 4. Cho một dãy các phần tử, mỗi phần tử được biểu diễn bằng một số tự nhiên. Hãy liệt kê tất cả các hoán vị của dãy đã cho trong hai trường hợp sau:

- Các phần tử khác nhau đôi một;
- Các phần tử có thể trùng nhau.

Yêu cầu: Xây dựng lớp Permutations2 có thuộc tính là một dãy số nguyên dương, và có phương thức `ArrayList<ArrayList<Integer>> listPermutations()` trả lại danh sách các hoán vị.

Ví dụ:

Đầu vào:

1 2 3

Đầu ra:

Co 6 hoan vi la:

[1 2 3]

[1 3 2]

[2 1 3]

[2 3 1]

[3 1 2]

[3 2 1]