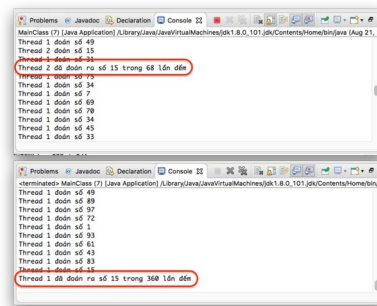


Bài 1. Hãy tạo ra một trò chơi để người dùng có thể nhập vào một số nguyên trong khoảng từ 1 đến 100. Sau đó bạn xây dựng một Thread đoán số, Thread này sẽ đạo ra các con số random cũng trong khoảng 1 đến 100 đó. Cứ mỗi lần random được một số, Thread sẽ in ra console cho người chơi có thể nhìn thấy. Thread sẽ dừng lại khi random ra một số trùng với số mà người chơi vừa nhập, đồng thời in ra số lần “đoán” để ra được con số đó.

Lưu ý rằng, có n Thread cùng đoán số, để “thi thố” xem Thread nào “đoán” ra con số của người chơi nhanh nhất.

Để dễ hình dung hơn, hình ảnh dưới đưa ra kết quả console dự kiến của trò chơi. Kết quả này dựa trên việc “đoán” của 2 Thread, để có thể biết được người chơi đã nhập vào con số 15. Và như hình thì Thread 2 đã thắng với 68 lần đoán. Thread 1 “kém thông minh” hơn và tiếp tục đoán cho đến lần đoán thứ 360.



Bài 2: Tạo ra 2 tuyến thực hiện các nhiệm vụ sau:

- Tuyến 1: Sinh ra 1 số nguyên ngẫu nhiên trong khoảng từ 1 đến 20.
- Tuyến 2: Lấy số ngẫu nhiên mà Tuyến 1 sinh ra rồi tính bình phương của nó và hiển thị ra màn hình.

Chú ý: Cần thực hiện đồng bộ hóa để dữ liệu Tuyến 2 đã đọc số ngẫu nhiên do Tuyến 1 sinh ra rồi thì không đọc lại nữa. Tuyến 1 sinh ra số ngẫu nhiên, số này chưa được đọc bởi Tuyến 2 thì Tuyến 1 cũng không được sinh ra số ngẫu nhiên mới.

Viết hàm main minh họa.

Bài 3: Cho một danh sách là Mã sinh viên

Ví dụ: {"A12345","A22334","A13245","A00776","A54321"}

Một danh sách khác là Họ và tên tương ứng với các mã sinh viên ở trên.

Ví dụ: {"Nguyen Ngoc Lan","Duong Tien Nam","Ngo Anh Quan","Pham Van Duc","Tran Thi Ha"}

Viết 1 ứng dụng Java sử dụng 2 tuyến thực hiện công việc sau:

- Tuyến 1: Sau mỗi giây sẽ hiển thị mã sinh viên tương ứng trong danh sách tên
- Tuyến 2: Ngay sau đó sẽ hiển thị Họ và tên tương ứng của Mã sinh viên đó
- Hai tuyến thực hiện mỗi tuyến 5 lần rồi mới dừng lại và phải đảm bảo rằng tuyến 1 luôn chạy trước rồi mới đến tuyến 2.