

## (OPTIONAL and Challenging) Nim game (exercise below)

*Nim* is a classical game going back to antiquity. The game starts with a number of sticks (or pebbles) in two or more different piles. Two players play the game, taking turns to remove a number of sticks from one of the piles. The players can remove as many sticks from a pile as they want, but only from the same pile. The player that removes the last stick loses the game.

The following is from a computer program implementing the game.

```
||| | | | |
|||||
|||||||
Player 1, Choose a pile.
1
How many sticks do you want to remove?
2
|
|||||
|||||||
Player 2, Choose a pile.
3
How many sticks do you want to remove?
8
|
|||||

Player 1, Choose a pile.
4
Please pick a pile number between 1 and 3 that has sticks in it.
3
Please pick a pile number between 1 and 3 that has sticks in it.
2
How many sticks do you want to remove?
4
|
|

Player 2, Choose a pile.
2
How many sticks do you want to remove?
1
|

Player 1, Choose a pile.
1
How many sticks do you want to remove?
1

Player 2 wins
```

Your task is to write this program (see below)

## (Nim Game – Print the piles)

Create a class, `NimGame`, with a main method in which you

- Create a variable, `piles`, that represents the piles of sticks. Initialize it with 3 piles of 3, 5 and 8 sticks, respectively. (*Hint*: use an array to represent the number of sticks in each pile).

- b. Loop through the array and print the sticks to the console as shown above.

### **(Nim Game – Remove sticks from a pile)**

Make the following changes to the main method from the previous exercise:

- a. Ask the user for a pile. Keep asking until the user enters a valid number of a pile with sticks in it (a validation loop).
- b. Ask the user for a number of sticks. Keep asking until the user enters a valid number of sticks (at least 1, not more than are in the pile).
- c. Remove the stick and print the pile again.

### **(Nim Game – Play the game)**

Make the following changes to the main method from the previous exercise:

- a. Keep track on how many sticks are in the game.
- b. Keep making moves until there are no more sticks left.
- c. Keep track of player numbers.
- d. Announce the winner.

### **(Nim Game – Additions to the game)**

Make the following changes to the main method from the previous exercise:

- a. Change your game to start with a random number of piles (at most 5), and a random number of sticks in each pile (at most 10).

*Hint:* The following code randomly initializes `r` to 0, 1, 2 or 3:

```
Random random = new Random();  
int r = random.nextInt(4);
```

(You have to include `java.util.Random`)

- b. When a game is ended ask the players if they want to play a new game. If they do, ask them if they want the same start or a new random start. Then start the game again according to their choice