Stack and Queue

Stack е линейна структура от данни на която можем да достъпваме само последния добавен елемент или иначе казано (First in last out – FILO). Първият елемент, който бъде добавен ще може да бъде достъпен чак като последен. Например:

Имаме 1 2 3 4 5 за да стигнем до 4 първо трябва 5 да го няма и така до 1.

На Stack може да се добавя само от единия край, не може да се вмъква или достъпва нещо, което не е на краят му.

Трите основни команди са:

Push – добавяне на стойност към края на stack-a. Добавената стойност става последната в реда.

Pop – връща ни като резултат последната стойност в реда на Stack-a и я премахва от Stack-a.

Peek – проверява последната стойност в реда на Stack-a (връща без да премахва).

Синтаксис:

Създаване на празен Stack:

ArrayDeque <Integer(Тип данни)> stack(Име) = new ArrayDeque<>();

Добавяне на последен елемент:

Stack.push(елемент);

Премахване на последен елемент:

Integer element = stack.pop();

Проверка на последен елемент:

Integer element = stack.peek();

!Проверка за големина на: int size = stack.size();

!Проверка дали е празен: Boolean isEmpty = stack.isEmpty();

!Проверка дали съдържа дадена стойност: Boolean isExisting = stack.contains(пров. елемент);

!aко искаме да добавим цял масив например в стак то това става чрез следния метод:

Collections.addAll(Stack(име на стак-а), име на масива);

Queue е същото като Stack с разликата, че добавяме в края а вземаме от началото. Иначе казано при добавяне добавяната стойност става последна като при Stack, но когато вземаме се взема винаги първата добавена стойност, а НЕ последната.

Синтаксис:

ArrayDeque<Integer> queue(name) = new ArrayDeque<>();

Има 2 начина за добавяне:

queue.add() – throws exception if the queue is full

queue.offer() – returns false if the queue is full

Има 2 начина за премахване:

queue.remove() – throws exception if the queue is empty

queue.poll() – returns null if the queue is full, а ако не е празна премахва и връща стойността.

Проверка:

Queue.peek() – връща без да премахва, като при стаковете.

!Съществува и опашка със приоритет където може да се подреждат по приоритет но засега няма да я използваме!