

طراحی پشته

مشاهده‌ی اسلایدهای شش برای این تمرین الزامی است.

درمورد ساختمان داده پشته (stack) تحقیق کنید و کلاس پشته را با توجه به توابع زیر پیاده سازی کنید. تضمین می‌شود که بیش‌تر از صد عضو به پشته اضافه نشود. استفاده از کلاس Stack جاوا مجاز نیست.

```
1 | public boolean empty()
```

اگر پشته خالی باشد، مقدار true بر می‌گرداند.

```
1 | public int size()
```

این متد اندازه‌ی پشته را بر می‌گرداند.

```
1 | public String push(String element)
```

این متد عنصر element را به پشته اضافه می‌کند و همان را برمی‌گرداند.

```
1 | public String pop()
```

این متد عنصر سر پشته را برمی‌گرداند و آن را حذف می‌کند. اگر پشته خالی بود، رشته "error" را برمی‌گرداند.

```
1 | public String peek()
```

این متد عنصر سر پشته را بدون حذف‌کردن برمی‌گرداند. اگر پشته خالی بود، رشته "error" را برمی‌گرداند.

```
1 | public int search(String element)
```

به دنبال عنصر element می‌گردد. اگر آن را یافت اندیشش را برمی‌گرداند و در غیر این صورت 1- را برمی‌گرداند. اندیس را برای اولین عضو اضافه شده صفر، برای دومین عضو اضافه شده یک و به همین ترتیب برای آخرین عضو اضافه شده به اندازه یکی کمتر از اندازه پشته در نظر بگیرید.

```
1 | public void clear()
```

پشته را به صورت کامل خالی می‌کند.

```
1 | public String toString()
```

عناصر پشته را به شکل رشته و به فرمی که در مثال‌ها مشاهده خواهید کرد، برمی‌گرداند.

```
1 | public void pushAll(String... elements)
```

به تعداد دلخواه کاربر عنصر می‌گیرد و آنها را به ترتیب به انتهای لیست اضافه می‌کند.

موضوع varargs را در اسلاید هشت بخوانید. (می‌توانید بعد از خواندن این موضوع، این متد را پیاده‌سازی کنید.)

مثال

```
1 | SimpleStack stack = new SimpleStack();
2 | System.out.println(stack.empty()); // true
3 | stack.push("1");
4 | System.out.println(stack.toString().equals("{1}")); // true
5 | for (int i = 2; i <= 5; i++) {
6 |     stack.push(String.valueOf(i));
7 | }
8 | System.out.println(stack.toString().equals("{1, 2, 3, 4, 5}"));
9 | System.out.println(stack.peek()); // 5
10 | System.out.println(stack.pop()); // 5
11 | System.out.println(stack.pop()); // 4
12 | System.out.println(stack.toString()); // {1, 2, 3}
13 | System.out.println(stack.empty()); // false
14 |
```

```
System.out.println(stack.size()); // 3
System.out.println(stack.search("3")); // 2
System.out.println(stack.search("5")); // -1
stack.clear();
System.out.println(stack.toString()); // {}
System.out.println(stack.size()); // 0
System.out.println(stack.empty()); // true
System.out.println(stack.pop()); // error
System.out.println(stack.peek()); // error

// requirement: Slide 8
stack.pushAll("Ali", "ali", "aLi", "aLI", "ALI");
System.out.println(stack.toString()); // {Ali, ali, aLi, aLI, ALI}
System.out.println(stack.peek().equals("Ali")); // false
System.out.println(stack.peek().equals("aLi")); // false
System.out.println(stack.peek().equals("ALI")); // true
```

آن چه که باید آپلود کنید

آن چه که باید آپلود کنید، یک فایل zip است که وقتی آن را باز می‌کنیم، در آن فقط فایل SimpleStack.java را می‌بینیم.