در تمامی سؤالها، اگر نام یک کلاس یا تابع را فراموش کردید، یک نام دلخواه برای آن در نظر بگیرید و توضیح کوتاهی درباره آن ذکر کنید.

۱. متد زیر برای آرایه‌های بزرگ کند است. این متد را طوری تغییر دهید که در حداکثر سه Thread اجرا شود. (۱۰)

```
public static int numberOfSeyyedRezas(List<String> names){
	if(names==null)
		return 0;
	int num = 0;
	for (String s : names) {
		if(s==null)
			continue;
		String name = s.toLowerCase();
		if(name.startsWith("seyyed") && name.contains("reza"))
			num++;
	}
	return num;
}
```

۲. می‌دانیم test-case زیر با موفقیت اجرا می‌شود. نمودار UML Class Diagram را برای این کلاس‌ها طراحی کنید و این طراحی را به صورت کامل پیاده‌سازی کنید. (۲۵)

```
@Test
public void someTestCases(){
  Namable lionelMessi = new FootballPlayer("Lionel Messi");
  assertEquals(lionelMessi.getName(), "Lionel Messi");
  Person person = (Person) lionelMessi;

  Book hafezBook = new Book("Hafez", 10);
  assertTrue(hafezBook instanceof Namable);
  assertTrue(hafezBook instanceof HasPrice);
  assertTrue(lionelMessi instanceof HasPrice);
  assertFalse(new Person("Ali Alavi") instanceof HasPrice);

  person.buy(hafezBook);
  person.buy(new Book("Refactoring", 30));
  assertTrue(person.hasBought(new Book("Refactoring", 40)));
  assertFalse(person.hasBought(new Book("Harry Potter", 30)));
}
```

یادآوری: تابع assertTrue چک می‌کند که پارامتر آن مقدار true برگرداند، در غیر این صورت اعلام خطا می‌کند. تابع asseertFalse چک می‌کند که پارامتر آن مقدار false باشد، تابع assertEquals مراقب است که دو پارامتر آن با هم برابر باشند، تابع fail() هم همواره اعلام خطا می‌کند.

۳. می‌دانیم که یک شیء (Object) در یک فایل serialize شده است. درباره این شیء و کلاس آن چیزی نمی‌دانیم. متدی بنویسید که اسم یک فایل را به عنوان ورودی بگیرد و شیء را از فایل load کند و نام و مقدار فیلدهای آن را بنویسد. (۱۵)

۴. همان‌طور می‌دانید refactoring تغییر در ساختار و طراحی برنامه است، بدون این که کارکرد آن تغییر کند. کد زیر را refactor کنید تا ساختار بهتری پیدا کند. برای این کار از چه تکنیک‌هایی استفاده می‌کنید؟ (۱۵)

```
import java.util.Date;
enum MemberType { Student, Teacher, Guest}
class GUI {
  public void showPersonInfo(String firstName,
      String lastName, String nationalID,
      Date birthDate, MemberType type) {
    if (firstName == null || lastName == null
        || nationalID == null)
      return;
    if (firstName.isEmpty() || lastName.isEmpty()
        || nationalID.isEmpty())
      return;
    if (!firstName.matches("[a-z]+")
        || !lastName.matches("[a-z]+")
        || !nationalID.matches("[0-9]+"))
      return;
    showPerson(firstName, lastName, nationalID,
        birthDate, type);
  }
}
class Library {
  public Boolean addNewMember(String firstName,
      String lastName, String nationalID,
      Date birthDate, MemberType type) {
    if (firstName == null || lastName == null
        || nationalID == null)
      return false;
    if (firstName.isEmpty() || lastName.isEmpty()
        || nationalID.isEmpty())
      return false;
    if (!firstName.matches("[a-z]+")
        || !lastName.matches("[a-z]+")
        || !nationalID.matches("[0-9]+"))
      return false;
    return saveToDatabase(firstName, lastName,
        nationalID, birthDate, type);
  }
}
```

۵. واسط Stack را پیاده‌سازی کنید. (۱۵)

```
interface Stack<E>{
     void push(E e);
     E pop()throws StackEmptyException;
     E top()throws StackEmptyException;
     void clear();
}
```

پشته (Stack) ساختمان داده‌ای برای نگهداری اشیاء است. مانند پشته‌ای از بشقاب‌ها که روی هم قرار می‌گیرند. با کمک تابع push یک شیء روی پشته اضافه می‌شود، با کمک تابع pop از روی پشته برداشته می‌شود و با کمک top بدون حذف از روی پشته، شیءی که بالای پشته قرار گرفته برگردانده می‌شود. تابع clear هم پشته را خالی می‌کند.

پیاده‌سازی شما باید طوری انجام شود که test-case زیر با موفقیت انجام شود. همچنین از کلاس‌های Container مانند Collection، List و سایر کلاس‌های بسته java.util نباید استفاده کنید. هر چه لازم است، خودتان پیاده‌سازی کنید.

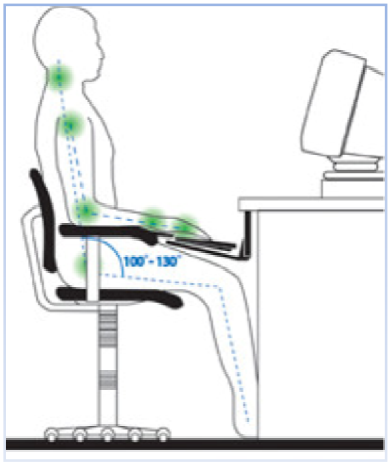
```
@Test
public void someTests() throws StackEmptyException {
	Stack<String> stack = new StackImpl<String>();
	try {
		stack.top();
		fail();
	} catch (StackEmptyException e) {}
	stack.push("ali");
	stack.push("gholi");
	assertEquals(stack.pop(), "gholi");
	assertEquals(stack.top(), "ali");
	assertEquals(stack.pop(), "ali");
	try {
		stack.top();
		fail();
	} catch (StackEmptyException e) {
	}
	stack.push("ghamar");
	stack.clear();
	try {
		stack.top();
		fail();
	} catch (StackEmptyException e) {}
}
```

۶. خروجی برنامه زیر چیست؟ (۱۵)

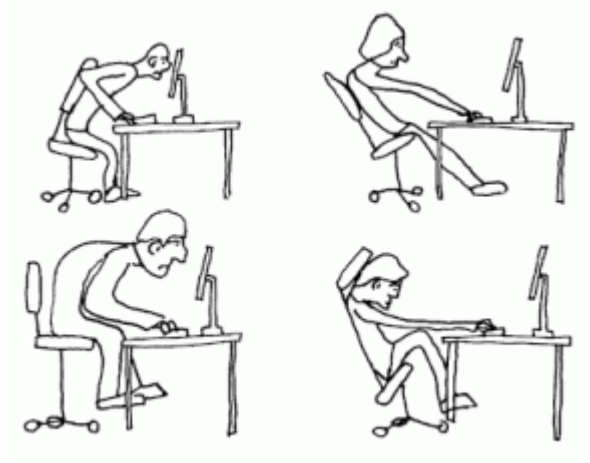
```
interface A{
	void a(A a);
}
abstract class B implements A{
	static String field1 = getDefaultString();
	public B(){
		System.out.println("Default Constructor in B");
	}
	private static String getDefaultString() {
		System.out.println("getDefaultString()");
		return "default";
	}
	public void a(B b){

	}
}
```

```java
class C extends B{
	static{
		System.out.println("Static block in C");
	}
	public C(){
		System.out.println("Default Constructor in C");
	}
	public C(String s){
		System.out.println("Constructor(String) in C");
	}
	public void a(A a) {
		System.out.println("a(A a) in C");
	}
	public void a(B b) {
		System.out.println("a(B b) in C");
	}
	public void a(C c) {
		System.out.println("a(C c) in C");
	}
}
public class WhatsOutput {
	public static void main(String[] args) {
		A a = new C();
		a.a(a);
		B b = new C("something");
		b.a(b);
	}
}
```

۷. در هنگام برنامه‌نویسی جاوا، نحوه درست نشستن روی صندلی کدام گزینه است؟ (۵)



گزینه الف



گزینه‌های ب

مؤمن همواره خانواده خود را از دانش و ادب شایسته بهره‌مند می‌سازد تا همه آنان را وارد بهشت کند. امام صادق(ع)

موفق باشید

علی‌اکبری