

برنام خالق یکتا

انجمن جاواکاپ

آزمون برنامه نویسی جاوا



*Sava Cup*  
association

نمونه آزمون جاوا: بخش پایه و حرفه‌ای

	مدت زمان پاسخگویی		تعداد سوالات
		نام و نام خانوادگی:	
		شماره داوطلبی:	

## سوالات بخش پایه

(۱) خروجی قطعه برنامه زیر چیست؟

```
String text = "Ali#and#Taghi#are#friends";
String[] array = text.replace('#', '!').split("!");
int sum=0;
for (String s : array)
    sum+=s.length();
System.out.println(sum);
```

1) 5	2) 9	3) 21
4) Ali and Taghi are friends	5) 4	6) 3

Sa\_jse\_fun\_xxx\_063

(۲) فرض کنید A و B و C و D به صورت زیر تعریف شده باشند:

```
interface A{}
interface B {void f();}
class C implements A{public void f(){} }
class D extends C implements A,B{}
```

همچنین فرض کنید متغیرهای a و b و c و d هم به صورت زیر تعریف شده باشند:

```
A a = null;
B b = null;
C c = null;
D d = null;
```

کدام گزینه‌های زیر دچار خطای کامپایل می‌شوند (سه گزینه را انتخاب کنید)؟

1) a.f();	4) d.f();	7) a=d;
2) b.f();	5) a=b;	8) b=c;
3) c.f();	6) a=c;	9) c=d;

Sa\_jse\_fun\_xxx\_005

(۳) کدام گزینه درباره برنامه زیر صحیح است؟

```
public class Recursive {
    static double f(int num) {
        return 1 + f(num - 1);
    }
    public static void main(String[] args) {
        System.out.println(f(1));
    }
}
```

- ۱- خطای کامپایل در خط `return 1 + f(num - 1);`
- ۲- خطای کامپایل در خط `System.out.println(f(1));`
- ۳- خطای زمان اجرا از نوع `StackOverflowError`
- ۴- خطای زمان اجرا از نوع `ArithmeticException`
- ۵- بدون خطا اجرا می‌شود ولی در یک حلقه بینهایت باقی می‌ماند (هرگز این برنامه تمام نمی‌شود)
- ۶- بدون خطا اجرا می‌شود و مقدار ۱ را چاپ می‌کند

Sa\_jse\_fun\_xxx\_083

(۴) به نظر شما قطعه برنامه زیر با چه هدفی نوشته شده است؟

```
a = s.trim().matches(".*@.*\\.\\.+.");
```

۱. رشته S را در متغیر a کپی می‌کند و سپس S را خالی می‌کند.
۲. اگر رشته S حاوی یک آدرس ایمیل باشد، مقدار a برابر با true خواهد شد.
۳. برخی کاراکترهای مشخص شده را از رشته‌ی S حذف می‌کند.
۴. تعداد نقطه‌های موجود در رشته S را در متغیر a ذخیره می‌کند.
۵. مجموع تعداد نقطه‌ها و تعداد backslash های رشته S را در متغیر a ذخیره می‌کند.
۶. با توجه به نقطه‌های موجود در رشته S، آن را به چند کلمه و عبارت مستقل تقسیم می‌کند.

Sa\_jse\_fun\_xxx\_031

(۵) می‌دانیم ArithmeticException زیرکلاس RuntimeException است (ArithmeticException extends RuntimeException)

خروجی برنامه زیر چیست؟

- 1) a b c d e
- 2) c d b
- 3) a b c
- 4) c b
- 5) c d e
- 6) a
- 7) a b
- 8) Exception
- 9) c d d e
- 10) c d d

```
public class Exceptions {
    static String s = "";
    public static void main(String[] args) {
        try {
            callMe();
            s += "a ";
        } catch (Exception ex) {
            s += "b ";
        }
        System.out.println(s);
    }
    static void callMe() throws RuntimeException {
        try {
            s += "c ";
            throw new ArithmeticException();
        } catch (ArithmeticException ie) {
            s += "d ";
        }
        callMeToo();
        s += "e ";
        throw new RuntimeException();
    }
    static void callMeToo() throws ArithmeticException {
        throw new ArithmeticException();
    }
}
```

Me\_jse\_fun\_xxx\_065

۶) کدام گزینه‌ها صحیح هستند؟ (۲ گزینه صحیح)

- ۱- حاصل عبارت `Object instanceof "Cat"` برابر با `true` است.
- ۲- حاصل عبارت `Cat instanceof "Cat"` برابر با `true` است.
- ۳- انقیاد پویا (`dynamic binding`) برای متدهای استاتیک رخ می‌دهد.
- ۴- سربار کردن متد (`method overloading`) برای متدهای استاتیک ممکن است.
- ۵- متدهای استاتیک را می‌توانیم `override` کنیم و این کار را با `@Override` تصریح کنیم.
- ۶- اگر یک متد `protected` باشد نمی‌تواند `abstract` باشد.

Sa\_jse\_fun\_xxx\_081

۷) کدام گزینه خروجی قطعه برنامه زیر را نشان می‌دهد؟

```
import java.util.HashSet;
import java.util.Set;
enum Color{GREEN, WHITE, RED}
public class Program {
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public static void main(String[] args) {
        Set set = new HashSet();
        set.add(Color.RED);
        set.add(Color.valueOf("RED"));
        set.add("RED");
        boolean allColors = true;
        for (Object object : set) {
            if(! (object instanceof Color))
                allColors = false;
        }
        System.out.println("" + allColors + set.size());
    }
}
```

- 1) true1
- 2) true2
- 3) true3
- 4) true4
- 5) false1
- 6) false2
- 7) false3
- 8) false

Sa\_jse\_fun\_xxx\_103

۸) کدام خط از برنامه زیر خطای کامپایل دارد؟ (شماره گزینه‌ها در بعضی از خطها به صورت کامنت آمده است)

```
public class OuterClass {
    private int value = 2; /*1*/
    class Inner{ /*2*/
        public void f(){ OuterClass.this.value = 5; /*3*/ }
    }
    static class InnerStatic{ /*4*/
        public void f(){ OuterClass.this.value = 5; /*5*/ }
    }
    public static void main(String[] args) {
        OuterClass outer = new OuterClass(); /*6*/
        OuterClass.Inner inner = outer.new Inner(); /*7*/
        InnerStatic innerStatic = new InnerStatic(); /*8*/
    }
}
```

Sa\_jse\_fun\_xxx\_082

```
import java.util.*;
class Human{
    int kindness;
    int money;
    public Human(int kindness, int money) {
        this.kindness = kindness;
        this.money = money;
    }
}
public class Containers {
    public static void main(String[] args) {
        List<Human> list = Arrays.asList(new Human(1, 10), new Human(5, 2));
        Comparator<Human> comparator = new Comparator<Human>(){
            @Override
            public int compare(Human o1, Human o2) {
                return o1.kindness<o2.kindness?-1:o1.kindness==o2.kindness?0:+1;
            }
        };
        System.out.println(Collections.max(list, comparator).money);
    }
}
```

- |      |      |      |      |      |       |      |
|------|------|------|------|------|-------|------|
| 1) 1 | 2) 2 | 3) 3 | 4) 4 | 5) 5 | 6) 10 | 7) 7 |
|------|------|------|------|------|-------|------|

Sa\_jse\_fun\_xxx\_078

```
class MyIterable implements Iterable<Integer>{
    Collection<Character> val;
    public MyIterable(Collection<Character> val) { this.val = val; }
    @Override
    public Iterator<Integer> iterator() {
        return Arrays.asList(1,2,3,3,4).iterator();
    }
}
public class MyProgram{
    public static void main(String[] args) {
        for (Object o : new MyIterable(Arrays.asList('A', 'A', 'B')))
            System.out.print(o);
    }
}
```

- |         |          |             |       |
|---------|----------|-------------|-------|
| 1- 1234 | 2- 12334 | 3- 4321     | 4- AB |
| 5- AAB  | 6- BAA   | 7- 12334AAB | 8- BA |

Sa\_jse\_fun\_xxx\_112

پایان سؤالات بخش پایه

## سوالات بخش حرفه‌ای

محتوای فایل Greeting.txt قبل از اجرای برنامه‌ی زیر، Salam است. پس از اجرای این برنامه، محتوای فایل چه خواهد بود؟

(۱۱)

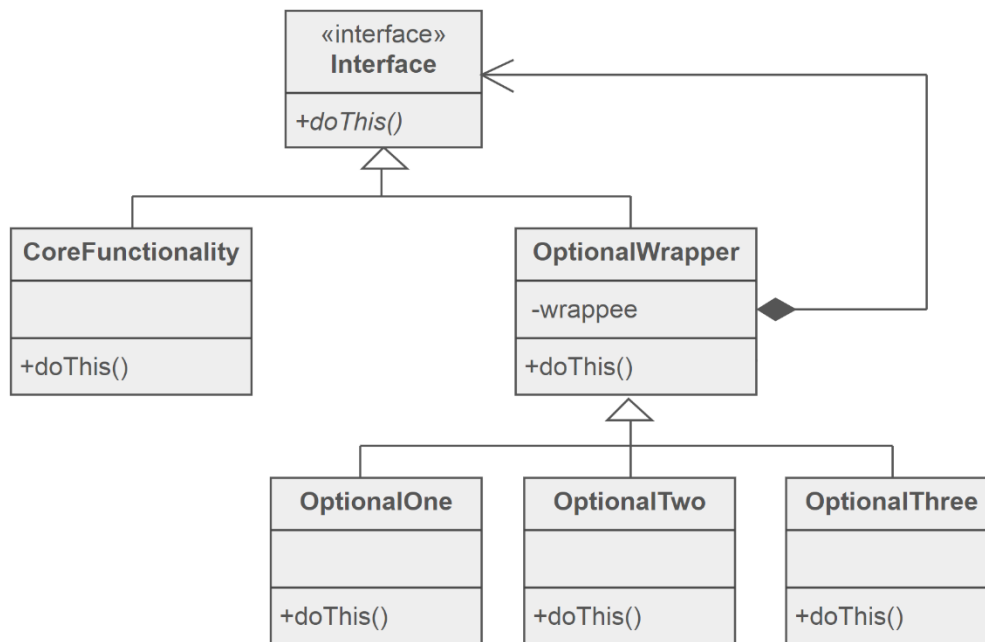
```
try (FileOutputStream out = new FileOutputStream("Greeting.txt", true)) {
    out.write("Bonjour".getBytes());
}
```

1- Salam	3- BonjourSalam	5- Bonjour
2- SalamBonjour	4- SalamSalam	6- فایل خالی خواهد شد

MK\_jse\_adv\_IO\_005

در نمودار UML زیر از کدام الگوی طراحی (Design Pattern) استفاده شده است؟

(۱۲)



1- Singleton	2- Builder	3- Proxy	4- Decorator
5- Factory	6- Adapter	7- Bridge	8- Prototype

MK\_jse\_adv\_OOD\_003

```

abstract class Shape {
    String color;
    protected abstract void draw();
}
class Circle extends Shape{
    Integer r;
    public Circle() {}
    public Circle(Integer r) { this.r = r; }
    public void draw() {}
}
public class Program {
    public static void main(String[] args) {
        Field[] cfs = Circle.class.getFields();
        Method[] cms = Circle.class.getMethods();
        Field[] sfs = Circle.class.getSuperclass().getFields();
        Method[] sms = Circle.class.getSuperclass().getMethods();
        System.out.println((cfs.length-sfs.length) * (cms.length-sms.length));
    }
}

```

1-	0	3-	-1	5-	-2	7-	-4
2-	1	4-	2	6-	4		

mf\_jse\_adv\_Reflection\_001

خروجی برنامه‌ی زیر کدام گزینه‌ها می‌تواند باشد؟ (دو گزینه صحیح)

```

public static void main(String[] args) throws InterruptedException {
    Thread[] threads = new Thread[5];
    for (int i = 0; i <= threads.length - 1; i++) {
        int ti = i+1;
        threads[i] = new Thread(() -> { System.out.print("T" + ti); });
    }
    for (int i = 0; i <= threads.length - 1; i++)
        if (i < 2) {
            threads[i].start();
            threads[i].join();
        } else
            threads[i].start();
}

```

1-	T1T2T3T4T5	3-	T5T4T3T2T1	5-	T5T3T4T1T2	7-	T3T4T5T1T2
2-	T2T1T3T5T4	4-	T2T1T3T4T5	6-	T3T5T4T2T1	8-	T1T2T5T4T3

mf\_jse\_adv\_Thread\_001

```
public class Book {
    String name, isbn;
    public Book(String name, String isbn) {
        this.name = name; this.isbn = isbn;
    }
    @Override
    public boolean equals(Object obj) {
        if (!(obj instanceof Book))
            return false;
        Book other = (Book) obj;
        if (isbn == null)
            return false;
        return isbn.equals(other.isbn);
    }
}
```

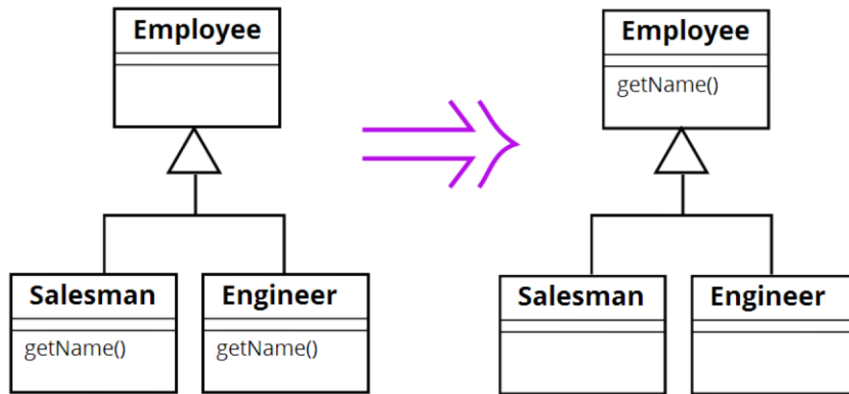
کدام گزینه در مورد تست کیس `testEquals` صحیح است؟

```
public class BookTest {
    Book book1, book2, book3, book4;
    @Before
    public void before() {
        book1 = new Book("A Man Called Ove", "12345");
        book2 = new Book("The Clown", "56789");
        book3 = new Book("Unknown Book", null);
        book4 = new Book("مردی به نام اوه", "12345");
    }
    @Test
    public void testEquals() {
        assertNotSame(book1, book4);
        assertNotEquals(book1, book2);
        assertNotEquals(book3, book4);
        assertEquals(book1, book4);
    }
}
```

- ۱- با موفقیت پاس می‌شود.
- ۲- در `assertNotSame(book1, book4)` با خطا مواجه می‌شود.
- ۳- در `assertNotEquals(book1, book2)` با خطا مواجه می‌شود.
- ۴- در `assertNotEquals(book3, book4)` با خطا مواجه می‌شود.
- ۵- در `assertEquals(book1, book4)` با خطا مواجه می‌شود.



نمودار زیر کدام تکنیک Refactoring را نشان می‌دهد؟ (۱۶)



- |                   |                             |                      |                       |
|-------------------|-----------------------------|----------------------|-----------------------|
| 1. Extract method | 3. Push down method         | 5. Extract interface | 7. Extract subclass   |
| 2. Pull up method | 4. Pull up constructor body | 6. Pull up field     | 8. Extract superclass |

MK\_jse\_adv\_Advanced\_005

کدام گزینه‌ها صحیح هستند؟ (دو گزینه صحیح) (۱۷)

- ۱- بازآرایی برنامه‌ها (code refactoring) وظیفه‌ی تیم تست است (وظیفه برنامه‌نویس نیست)
- ۲- باید بپذیریم که انجام بازآرایی (refactoring) در مجموع باعث کندی تولید و نگهداری نرم‌افزار می‌شود.
- ۳- ابزارهای تحلیل استاتیک کد (static code analysis) می‌توانند به پیدا کردن bad smell ها کمک کنند (ابزارهایی مثل Checkstyle و PMD و FindBugs و SonarQube)
- ۴- هدف اصلی از انجام بازآرایی (refactoring) افزایش سرعت اجرای برنامه است.
- ۵- بازآرایی (refactoring) باعث افزایش خوانایی و بهبود ساختار برنامه‌ها می‌شود

Sa\_jse\_adv\_Advanced\_016

کدام یک از Comparator های زیر، رشته‌ها را تنها براساس طول آن‌ها مقایسه می‌کند؟ (۱۸)

- |    |   |
|----|---|
| 1- | <code>Comparator&lt;String&gt; c1=(str1, str2) -&gt; str1.length() - str2.length();</code>          |
| 2- | <code>Comparator&lt;String&gt; c2=(str1, str2) -&gt; str1.compareTo(str2);</code>                   |
| 3- | <code>Comparator&lt;String&gt; c3=(str1, str2) -&gt; str1.length() &gt; str2.length();</code>       |
| 4- | <code>Comparator&lt;String&gt; c4=(str1, str2) -&gt; str1.length().compareTo(str2.length());</code> |

MK\_jse\_adv\_Java8\_002

پایان سوالات بخش حرفه‌ای

