

۱. با توجه به سابقه خود به یکی از سوالات (الف) یا (ب)، اما نه هر دو، پاسخ دهید. (توضیح کافی: یک پاراگراف تا یک صفحه) (۱۵ نمره)
- الف) اگر تجربه حرفه ای دارید، سازمان شما در کدام سطح از بلوغ آزمون^۱ عمل می کند؟
- ب) اگر تا به حال هیچگونه تجربه حرفه ای ندارید، به نظرتان نگاه شما به آزمون در کدام سطح بلوغ آزمون قرار دارد؟

This is an introspective exercise without a "correct" answer, but with justification. The idea is to place ourselves (and possibly our company) into the context described by Beizer's classification.

۲. به نظر شما چه فاکتورهای در ارتقاء یک سازمان از سطح ۲ به سطح ۴، کمک می کند؟ توضیح دهید. (۱۵ نمره)

Testing level 2: testing is to show errors

Testing level 4: a mental discipline that increases quality

This question lends itself to open ended discussion. A few

- (a) Good, technical people in testing.
- (b) Amicable relationships between testers and developers.
- (c) Management leadership.

۳. هدف از این تمرین تأمل بیشتر درباره آزمون است. همچنین این تمرین ارتباط تنگاتنگ میان ابهام در مشخصات^۲، عیب^۳، و موارد آزمون^۴ کارآمد را نشان می دهد.

الف) یک متد جاوا با امضای^۵ زیر بنویسید. این متد باید یک Vector شامل اشیایی که در یکی از دو Vector ورودی هستند، بازگرداند. (۱۰ نمره)

```
public static Vector union (Vector a, Vector b)
```

ب) با کمی دقت، احتمالاً متوجه مشکلات و ابهامات تعریف متد، که احتمال وجود عیب را زیاد میکند، خواهید شد. هر تعداد عیب که به ذهنتان می رسد را مشخص کنید. (۱۵ نمره)

Many potential ambiguities exist. The method name suggest set union, but the argument types (i.e. Vector) suggest otherwise. What happens if an input vector is null? What happens if an input vector contains duplicates? What happens if an input vector contains null entries? Are the inputs modified? Is the return value always a new set, or might it be either a or b? The choice of argument types and return types as vector is suspect; best practice is to use the least specific type consistent with the specifier's goals. In particular, if vector is replaced with List, this solves some, but not all of the problems above. Replacing vector with Set is probably an even better idea, assuming the arguments are really intended to be sets.

- ج) مجموعه ای از آزمون ها طراحی کنید که به نظر شما با احتمال قابل قبولی عیب های شناسایی شده در قسمت قبل را آشکار می کند. دلیل وجود هر آزمون را مشخص کنید. مجموعه آزمون ها را بر روی برنامه خود اجرا کنید و نتیجه را نشان دهید. (۳۰ نمره)
- * برای نمایش کدها و نتیجه اجرای آزمون ها از تصویر صفحه^۶ استفاده کنید.

See the examples in the specification below for some sample test cases, as well as the corresponding rationale. There could be other and additional tests. Note: Other choices for resolving ambiguities are certainly possible. For example, the specifier could decide to treat `null` inputs as if they were simply empty sets. Or, the specifier could prohibit `null` inputs with a precondition. This latter option is typically, but not always, avoided, on the principle that preconditions are an invitation to misbehaviour by clients.

د) امضای متد را به گونه ای بازنویسی کنید که به اندازه کافی بدون ابهام و دقیق باشد. می توانید برای توضیح بیشتر از مثال هایی که از آزمون های قسمت قبل استخراج شده اند، کمک بگیرید. (۱۵ نمره + ۵؟)

Here, we have lots of options.

1. We may change the argument and return types to reflect ordinary set union. This change alone handles many of the ambiguities. The use of examples in specifications, while not strictly formal, is extremely helpful in conveying design decisions from the specifier to the reader.

```
public static Set union (Set a, Set b)
// EFFECTS: if a or b is null throw NullPointerException
// else return a new Set that is the set union of a, b
// e.g. union (null,{}) is NullPointerException
// e.g. union ({1,2},{2,3}) is {1,2,3}
// e.g. union ({1,2},{cat,hat}) is {1,2,cat,hat}
// e.g. union ({null},{cat,hat}) is {null,cat,hat}
// e.g. union (a,a) is Set t st that t!=a, but t.equals(a)
```

2. Introducing generics makes the solution safer. A straight-forward generified solution has the signature:

```
public static <E> Set<E> union (Set<E> a, Set<E> b)
```

3. A generified solution that properly respects the type hierarchy has the signature:

```
public static <E> Set<E> union (Set<? extends E> a, Set<? extends E> b)
```

What is interesting about generic solutions is that they transform run time errors, usually in the form of `ClassCastException`, to compile time errors. This is a good thing! The `union` example isn't really restrictive with respect to types, so this fact isn't well illustrated here. However, readers might consider the generic version of `max()`, which does illustrate this property. For example, try to call the following `max()` method with a list mixing `Integer` and `String` elements. If you have properly used generics, it won't be possible to complete the compilation since a `String` can't be compared to an `Integer`.

```
public static <T extends Comparable<? super T>> T max (List<? extends T> list)
```

- پاسخ تمرین ها به صورت تایپ شده، در قالب یک فایل Pdf، در مودل بارگزاری کنید.
 - نام فایل خود را مطابق الگوی زیر قرار دهید:
- ST-HW1-[FamilyName(s)].pdf
- پاسخ تمرین های تشریحی به زبان فارسی باشند.
 - لطفا نظم، ساختار و توالی سوالات را در پاسخ ها رعایت کنید.
 - به ازای هر روز تاخیر در تحویل تمرین، ۲۰٪ از نمره تمرین کسر خواهد شد.
 - در صورت تحویل گروهی، رعایت نکات مربوط در شیوه نامه درس ضروری است.