Worksheet 8: Generics

Updated: 11th May, 2018

1. Discussion: Bounded Generics

For each of the following:

- Explain what it means.
- Give an example of declaring a `MyClass` reference variable.

(a)
```
public class MyClass<S, T extends S, U extends T> {...}
```

(b)
```
public class MyClass<E, T extends List<E>> {...}
```

(c)
```
public class MyClass<E, T extends List<? extends E>> {...}
```

2. Discussion: Valid or Invalid?

For each of the following, use your reasoning, research and/or coding skills to figure out if it's valid or invalid in Java:

(a)
```
public class MyClass<T> extends T {...}
```

(b)
```
public <T extends Exception> void method() throws T {...}
```

(c)
```
public <T> T getArrayMiddle(T[] array) {...}
```

(d)
```
public class MyClass<T extends Collection>
{
    private T<String> container1;
    private T<LocalTime> container2;
    ...
}
```

3. Discussion: Intersection Types

The lecture notes give the following example of an intersection type:

```
public interface Encodable { String encode(); }
public interface Viewable { UIElement view(); }

public class EmailAttachment<T extends Encodable & Viewable>
{
    private String filename;
    private T attachedObject;
    ...
}
```

Here is a modified approach that *doesn't* use generics:

```
public interface Encodable { String encode(); }
public interface Viewable { UIElement view(); }

public interface EncodableViewable extends Encodable, Viewable {}

public class EmailAttachment
{
    private String filename;
    private EncodableViewable attachedObject;
    ...
}
// Note that the 'extends' clause here is valid. Interfaces can 'extend'
// any number of other interfaces.
```

What is the practical difference between the two approaches?

4. Discussion: CSV Parser

Say your team is writing a general-purpose CSV-parsing library, and one team member has started with the following code:

```
public abstract class CsvColumn<T> extends ArrayList<T>
{
    public abstract void parse(String value);
}
```

> **Note:** As you probably know, a CSV (comma separated value) file is a simple kind of spreadsheet file, representing information in rows and columns. Like other spreadsheets, data can be textual, numerical, or chronological (dates and times). Spreadsheets are typically arranged such that a given column stores a single kind of data.
>
> When you want to parse a CSV file, you typically know in advance what each column is for, and hence what kind of data it should contain.
>
> In practice, CSV files can also be made to contain spreadsheet formulae (e.g. "`=A1+B2`"), but you can ignore that…unless you'd like the extra challenge!

Given some educated guesses as to the design intention:

(a) Suggest what the subclasses are likely to be, and how they would be declared.

(b) Suggest what the significance of "`extends ArrayList<T>`" is, and what it means about how this class is used.

(c) Suggest how the `parse()` method implementations probably work, assuming the parameter represents the string value of *one cell* in the file.

(d) What problem could we run into if the file contains column headings in the first row?

5. Implementing Type Safety

Obtain a copy of SpaceProbe.zip. Sadly this doesn't contain the *complete* code for the next space mission. In fact, it's poorly written, using type-unsafe operations (i.e. down-casting).

Background: a space probe has three on-board resources it must keep track of, in this case: fuel, battery power, and sensors. As shown in the code:

- Fuel is a dual quantity – the amount of oxygen and the amount of hydrogen remaining.
- Battery charge is just a single number.
- Sensors are not a quantity at all. Rather, there is a set of distinct instruments aboard the probe. During the long voyage through space, the probe's sensors will tend to fail somewhat randomly, so that this resource diminishes over time.

These resources are all expected to become depleted over time, and the code contains methods for handling this, and determining how much remaining time each resource has.

Refactor the code to make it type-safe. Your modifications should be minimalistic.

End of Worksheet