

Base Language - Feature #7126

replace 'new unknown()' with 'unknown.INSTANCE' in FWD conversion and runtime

02/15/2023 08:09 AM - Constantin Asofiei

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		version:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 02/15/2023 08:11 AM - Constantin Asofiei

In 7026a rev 14488, new unknown() was replaced with unknown.INSTANCE in parts of FWD runtime which showed up in profiling a large customer application. We need to make this more general:

- for conversion, any new unknown() must be converted to unknown.INSTANCE
- for runtime, replace all new unknown() occurrences with unknown.INSTANCE

As an unknown instance is immutable, there is no reason to create new instances of it.

#2 - 02/15/2023 08:44 AM - Greg Shah

Good idea. My only critique is that I want to find a different name than unknown.INSTANCE. I think this is a bit confusing.

How about if we add import static com.goldencode.p2j.util.unknown.UNKNOWN; and then simply reference UNKNOWN? We could also call it UNKNOWN_VALUE but I'd prefer the shorter text.

#3 - 02/15/2023 08:47 AM - Constantin Asofiei

Greg Shah wrote:

How about if we add import static com.goldencode.p2j.util.unknown.UNKNOWN; and then simply reference UNKNOWN?

That works, too. I already have this unknown.INSTANCE name in 7026a, I'll rename it in 7026a. Btw, I'm not planing to work on this task at this time, as the other new unknown() references did not show in the profiler.

#4 - 02/15/2023 01:03 PM - Constantin Asofiei

Renamed unknown.INSTANCE to unknown.UNKNOWN in 7026a/14493.

#5 - 04/10/2023 10:02 AM - Vladimir Tsichevski

I can take this task.

A question: currently there is 3 ways to obtain the unknown value:

1. calling `new unknown()`, we want to get rid of this method, OK. To make sure, just make this constructor private.
2. referencing the `unknown.UNKNOWN` constant. I gather, this **is** the recommended method.
3. calling the `unknown.instantiateUnknown()`, this function just returns the `unknown.UNKNOWN` constant. This method is **called** 8 times in the package, never referenced as a lambda.

I think, the last method should be eliminated either?

#6 - 04/10/2023 10:45 AM - Constantin Asofiei

Vladimir Tsichevski wrote:

I can take this task.

Greg?

1. calling `new unknown()`, we want to get rid of this method, OK. To make sure, just make this constructor private.

Correct, this will enforce the constant usage.

1. referencing the `unknown.UNKNOWN` constant. I gather, this **is** the recommended method.

Correct.

1. calling the `unknown.instantiateUnknown()`, this function just returns the `unknown.UNKNOWN` constant. This method is **called** 8 times in the package, never referenced as a lambda.

This needs to remain, this is an impl for the abstract method in BDT.

Also, all "unknown" references in the TRPL code need to be reviewed and replaced with `unknown.UNKNOWN` instead of `new unknown()`.

#7 - 04/10/2023 10:49 AM - Vladimir Tsichevski

Constantin Asofiei wrote:

1. calling the `unknown.instantiateUnknown()`, this function just returns the `unknown.UNKNOWN` constant. This method is **called 8** times in the package, never referenced as a lambda.

This needs to remain, this is an impl for the abstract method in BDT.

Now I see it, I have not noticed this due to the missing `@Override` annotation :-)

#8 - 04/10/2023 10:53 AM - Greg Shah

This task isn't a priority right now.