Q1: What are the reasons of using generics here? What are the benefits? Discuss with your partner and elaborate.

The benefits of using generics are that they are not depended on one instance, basically meaning that they are dynamic. By using generics you get stronger type checks at compile time, fixing compile time errors is easier than fixing runtime errors.

Q2: Compile and analyze compiler output. Given that "everything is an Object" in Java what is the cause of the problem reported by the compiler, if any?

The cause of the first problem is that contravariance is not allowed in input parameters, Object < BankAccount. The first problem regarding the function setValue. The cause of the second problem is that baCls is created as an object and this object does not have the function deposit (reflection issue).

Q3: How does this affect the compilation process? What is the problem, if any? What does the myAccount variable hold when the code is executed? Discuss with your partner whether your diagnosis in Q2 was correct.

We minimize our compilation errors to one since the myAccount variable is ininiated as a BankAccount class. The issue we get from this is that the Java compiler thinks that baCls is initiated as an object class type, so when we call baCls.newInstance() we return an object which is not what we want.

Q4: What does the dynamic cast do here? Is it the compiler that performs the cast operation or the Java runtime environment (JVM)? Is this code safe*?

The dynamic cast solves the problem as it casts the baCsl.newInstance() into the right type. The cast is perfomed in the Java runtime enviroment. We can simply switch out the type BankAccount with the type String and it will compile. The code is not safe, since casting is not safe and we can cast baCl to whatever we like.

Q5: Explain the compiler output? Are there errors? What is the reason? What does it say about the role of generics?

There are no errors, since baCls is of class BankAccount.The baCls variable with the annotion will in compile time convert the generic class object to an class object with the tpye BankAccount.

Q6: What is the run-time output? Reason why you get such an output and how does this relate to generics and their use with reflective instantiation of objects.

The run-time output is "115.0" followed by "EQUAL". The reason to this is that generics are not reified at run-time. This means the information is not present at run-time.

Students
Ruben Das - rubda680
Rasmus Eriksson - raser891