Prof. O. Nierstrasz
Mohammadreza Hazhirpasand, Joel Niklaus

Solution Objects and Types

- Exercises are given every week on the PL page of the SCG website (http://scg.unibe.ch/teaching/pl)
- Solutions to each assignment must be sent to **mohammadreza.hazhirpasand@inf.unibe.ch**
- The solutions of the assignments are to be delivered before every Thursday at 11 PM. Solutions handed in later than the specified time will not be accepted. In case of serious reasons send an e-mail to **mohammadreza.hazhirpasand@inf.unibe.ch**

Exercise (6 points)

1. What is the difference between subtyping and subclassing? Provide an example for your explanation. (1.5 pts)

   Answer:

   Subtyping: B is a subtype of A if anywhere one can use an A, a B could also be used. Subtyping has substitution. B is a subtype of A if an object of B can masquerade as an object of A in any context. For instance, Square is not a true subtype of Rectangle because: (1) Rectangles are expected to have a width and height that can be mutated independently, (2) Squares violate that expectation

   Subclassing: Subclassing should not be confused with subtyping. as a whole, subtyping acknowledges an is-a relationship. In contrast, subclassing reuses exiting implementation, factors out repeated code, and establishes a syntactic relationship. In addition, inheritance does not ensure behavioral subtyping. Subclassing provides

2. Using the Java class-interface hierarchy given in Figure 1, explain what is the relationship between classes and interfaces. (1.5 pts)

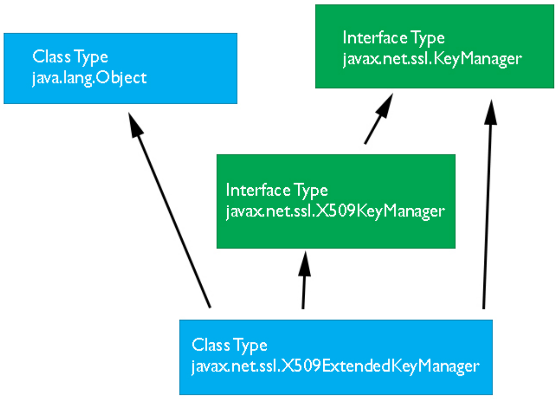

Figure 1: Java interface hierarchy

   Answer:

javax.net.ssl.X509ExtendedKeyManager is a subtype of java.lang.Object.

javax.net.ssl.X509ExtendedKeyManager is a subtype of javax.net.ssl.X509KeyManager.

javax.net.ssl.X509ExtendedKeyManager is a subtype of javax.net.ssl.KeyManager.

javax.net.ssl.KeyManager is a supertype of javax.net.ssl.X509ExtendedKeyManager.

javax.net.ssl.KeyManager is a supertype of javax.net.ssl.X509KeyManager.

java.lang.Object is a supertype of all other class and interface types.

3. Which forms of polymorphism are used in the Java code in Listing 1? Explain each of the forms. (1.5 pts)

```
public class Bern<TT> { // Hint 2
    private TT var1;
    public void set(TT mh) { this.var1 = mh; }
    public TT get() { return var1; }

 public static void main(String[] args) {
   int a = 3;
   float b = 2F;
   b = a;    // Hint 1
   System.out.println(b);
   Bern<Integer> mj = new Bern();
   mj.set(12);
   System.out.println(mj.get());
   }
}
```

Listing 1: Forms of polymorphism

Answer:

Hint 1: Coercion
Hint 2: Parametric Polymorphism

4. In the Java code Listing 2, explain what concept (covariance or contravariance) exists and why. (1.5 pts)

Answer:

Covariance

In Java, Arrays are covariant. an array of type T[] can store array of subtype S[], which means accepting subtypes.
In the example, the supertype (Number) can accept the subtype (Integer).

```java
Double[] mh = new Double[2];
   mh[0] = 100.2;
   mh[1] = 200.2;

   Number[] nm = mh;

   nm[0] = 1000.5;

   System.out.println(nm[0]);
```

Listing 2: Forms of polymorphism