

Theory: Referencing subclass objects

🕒 17 minutes 0 / 5 problems solved

Skip this topic

Start practicing

§1. How to refer to a subclass object

There are two ways to refer to a subclass object:

1. **using the subclass reference.** A subclass reference can be used to refer to its object;
2. **using the superclass reference.** A reference variable of the superclass can be used to refer to any subclass object derived from that superclass because a subclass is a special case of the superclass.

Let's consider an example of a class hierarchy.

```
1 class Person {
2
3     protected String name;
4     protected int yearOfBirth;
5     protected String address;
6
7     // public getters and setters for all fields
8 }
9
10
11 class Client extends Person {
12
13
14     protected String contractNumber;
15
16     protected boolean gold;
17
18
19     // public getters and setters for all fields
20 }
21
22
23 class Employee extends Person {
24
25
26     protected Date startDate;
27
28     protected Long salary;
29
30
31     // public getters and setters for all fields
32 }
33 }
```

As you know, each of the presented classes has the default no-args constructor.

Now let's see both approaches to the reference in action.

1. **Subclass reference.** We can create instances of the subclasses using the constructor:

```
1
Person person = new Person(); // the reference is Person, the object is Person
2
Client client = new Client(); // the reference is Client, the object is Client
3
Employee employee = new Employee(); // the reference is Employee, the object is Employee
```

Current topic:

[Referencing subclass objects](#) ...

Topic depends on:

✗ [Protected modifier](#) ...

Topic is required for:

[Polymorphism](#) ...

[The Object class](#) ...

[Runtime type checking](#) ...

[Encapsulating object creation](#) ...

In this case, we used **subclass references** because the types of references are the same as the types of created objects.

2. Superclass reference. When creating objects using the constructor, we can refer to a subclass object using the reference to the superclass:

```
1 | Person client = new Client(); // the reference is Person, the object is Client
2 | Person employee = new Employee(); // the reference is Person, the object is Employee
```

In this case, we used the superclass reference because references have the type of the superclass and the actual types of created objects are subclasses.

There are some points you need to remember:

- you cannot assign an object of one subclass to the reference of another subclass because they don't inherit each other:

```
1 | Client whoIsIt = new Employee(); // it's impossible
```

- you cannot assign an object of the parent class to the reference of its subclass:

```
1 | Client client = new Person(); // it's impossible too
```

The basic rule goes like this: *If class A is a superclass of class B and class B is a superclass of class C then a variable of the class A can reference any object derived from that class (for instance, objects of the class B and the class C). This is possible because each subclass object is an object of its superclass but not vice versa.*

§2. Accessing fields and methods through a superclass reference

We can use a superclass reference for any subclass object derived from it. However, we cannot access specific members of the subclass through the base class reference. We have access only to those members of the object defined by the type of reference.

Example. In the considered hierarchy, each class has getters and setters to access protected fields from the outside.

```
1 | Person employee = new Employee();
2 |
3 | employee.setName("Ginger R. Lee"); // Ok
4 | employee.setYearOfBirth(1980); // Ok
5 | employee.setSalary(30000); // Compile-time error, the base class "doesn't know" about the method
```

The superclass `Person` doesn't have the method `setSalary` of the class `Employee`. You cannot invoke the method through the superclass reference. The same rule works for fields as well.

§3. Casting between superclass and subclass

You can always cast an object of a subclass to its superclass. It may also be possible to cast an object from a superclass type to a subclass, but only if the object really is an instance of this subclass, otherwise `ClassCastException` will be thrown. Be careful when casting a class to its subclass.

```
1 Person person = new Client();
2
3 Client clientAgain = (Client) person; // it's ok
4
Employee employee = (Employee) person; // the ClassCastException occurs here
```

After successfully casting a superclass to a subclass, we can access its members.

§4. When to use the superclass reference?

When to use a superclass reference in practice may not be so obvious. Moreover, this adds some restrictions on access to class members. There are two common cases:

- processing an array (or another collection) of objects which have different types from the same hierarchy;
- a method that accepts an object of the base class, but can also work with objects of its subclasses.

Next, we combined both of these cases into a single example. Our method called `printNames` takes an array of `Person` and displays the names.

```
1 public static void printNames(Person[] persons) {
2     for (Person person : persons) {
3         System.out.println(person.getName());
4     }
5 }
```

This method will work for an array with `Person`, `Client` and `Employee` objects.

```
1 Person person = new Employee();
2 person.setName("Ginger R. Lee");
3
4 Client client = new Client();
5 client.setName("Pauline E. Morgan");
6
7 Employee employee = new Employee();
8 employee.setName("Lawrence V. Jones");
9
10
11 Person[] persons = {person, client, employee};
12
13 printNames(persons);
```

The output is exactly as we expected:

```
1 Ginger R. Lee
2 Pauline E. Morgan
3 Lawrence V. Jones
```

Thus, base class references have applications in some practical cases. Other cases of using the superclass references will be considered in topics related to **polymorphism**.

Table of contents:

[1 Referencing subclass objects](#)

[§1. How to refer to a subclass object](#)

[§2. Accessing fields and methods through a superclass reference](#)

[§3. Casting between superclass and subclass](#)

[§4. When to use the superclass reference?](#)

[Feedback & Comments](#)

 Report a typo

434 users liked this theory. 30 didn't like it. What about you?



Start practicing

[Comments \(20\)](#)

[Hints \(0\)](#)

[Useful links \(1\)](#)

[Show discussion](#)