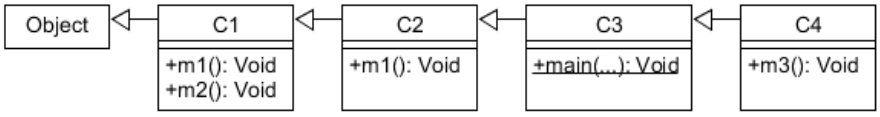
i219 Software Design Methodology
Sample Answers of Assignment 5



Describe how to deal with the message passing:

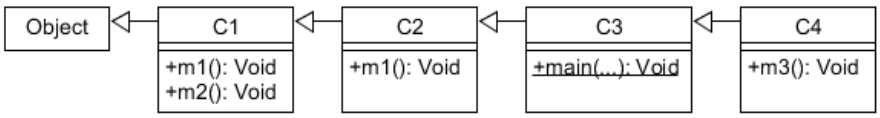
(new C4()).m1();

1. Compile time

The type information of the receiver object (new C4()) available at compile time is C4 and then C4 and all of its super classes are the scope from which we gather all candidate methods *name*(T_1,...,T_n), where each T_i is a type, such that *name* is m1 and *n* is 0, namely that there is no parameter. m1() is the candidate because it appears in C1 and C2. Since there is one and only one candidate, it is the minimum. Therefore, m1() is the method signature given to the message passing.

2. Runtime

The type information of the receiver object (new C4()) available at runtime is C4 and then C4 is the starting place where the search for the method invoked starts. Since m1() in C2 is the method whose signature is the same as the one given to the message passing and the nearest to the starting place in C4 and all of its superclasses, it is invoked for the message passing.



Describe how to deal with the message passing:

```
(new C4()).m1();
```

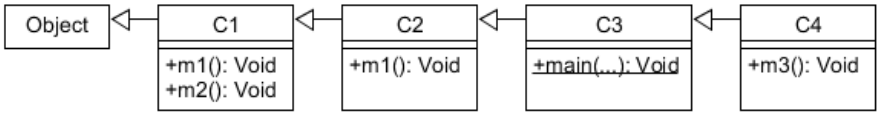
1. Compile time

The type info of (new C4()): C4
The type info of the parameter (): no parameter
The scope: C4 and all of its super
The candidates: m1()
The method signature given: m1()

2. Runtime

The type info of (new C4()): C4 (the starting place)
The method invoked: m1() in C2

You could use this abbreviation style.



Describe how to deal with the message passing:

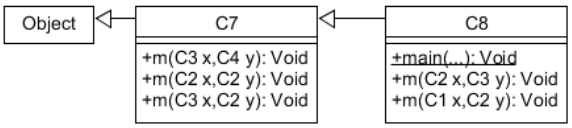
```
C1 o = new C4();
o.m3();
```

1. Compile time

The type info of o: C1
The type info of the parameter (): no parameter
The scope: C1 and all of its super
The candidates: none
The method signature given: none because of no candidate
Therefore, compiler error occurs.

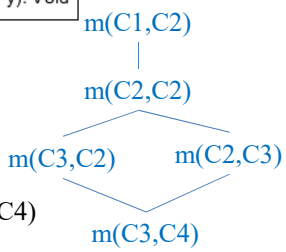
2. Runtime

It cannot be run because it cannot be compiled.



Describe how to deal with the message passing:

(new C8()).m(new C4(),new C4());

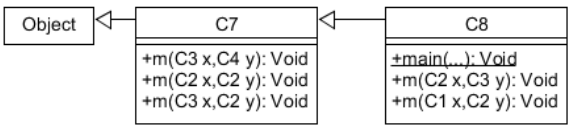


1. Compile time

The type info of (new C8()): C8
The type info of the parameter (new C4(),new C4()): (C4, C4)
The scope: C8 and all of its super
The candidates: m(C3,C4), m(C2,C2), m(C3,C2), m(C2,C3), m(C1,C2)
The method signature given: m(C3,C4) because it is the minimum element

2. Runtime

The type info of (new C8()): C8 (the starting place)
The method invoked: m(C3 x, C4 y) in C7

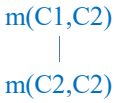


Describe how to deal with the message passing:

C2 x = new C4(); C2 y = new C4();
(new C8()).m(x,y);

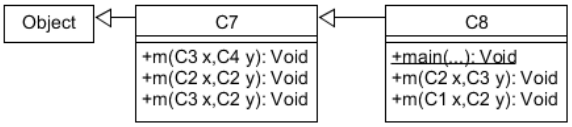
1. Compile time

The type info of (new C8()): C8
The type info of the parameter (x, y): (C2, C2)
The scope: C8 and all of its super
The candidates: m(C2,C2), m(C1,C2)
The method signature given: m(C2,C2) because it is the minimum element



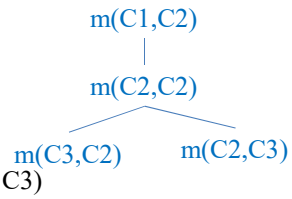
2. Runtime

The type info of (new C8()): C8 (the starting place)
The method invoked: m(C2 x, C2 y) in C7



Describe how to deal with the message passing:

(new C8()).m(new C3(),new C3());

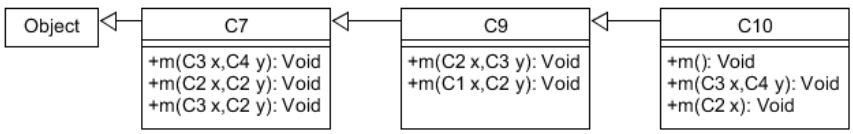


1. Compile time

The type info of (new C8()): C8
The type info of the parameter (new C3(),new C3()): (C3, C3)
The scope: C8 and all of its super
The candidates: m(C2,C2), m(C3,C2), m(C2,C3), m(C1,C2)
The method signature given: none because there is no minimum element
Therefore, compiler error occurs.

2. Runtime

It cannot be run because it cannot be compiled.



Describe how to deal with the message passing:

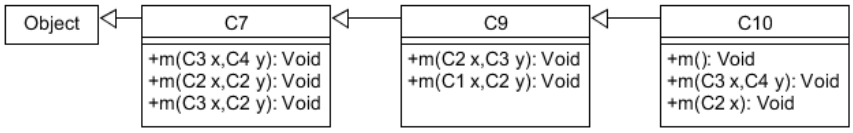
(new C10()).m();

1. Compile time

The type info of (new C10()): C10
The type info of the parameter (): no parameter
The scope: C10 and all of its super
The candidates: m()
The method signature given: m()

2. Runtime

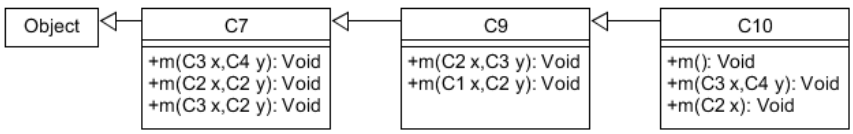
The type info of (new C10()): C10 (the starting place)
The method invoked: m() in C10



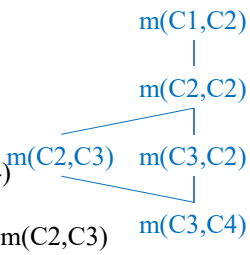
Let m() in C10 be as follows:

```
public void m() {
   this.m(new C4(),new C4());
   super.m(new C4(),new C4());
   this.m(new C3());
   super.m(new C3());
}
```

When m() in C10 is invoked for (new C10()).m();, describe how to deal with the four message passings there.



this.m(new C4(),new C4());

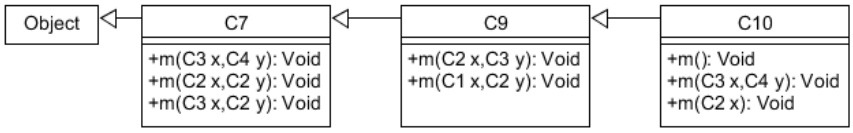


1. Compile time

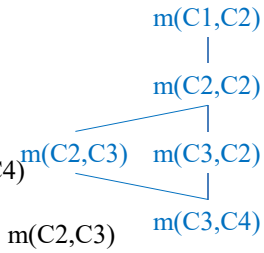
The type info of this: C10
The type info of the parameter (new C4(), new C4()): (C4, C4)
The scope: C10 and all of its super
The candidates: m(C3,C4), m(C2,C2), m(C3,C2), m(C1,C2) , m(C2,C3)
The method signature given: m(C3,C4) because it is the minimum element.

2. Runtime

The type info of this: C10 (the starting place)
The method invoked: m(C3 x, C4 y) in C10



super.m(new C4(),new C4());

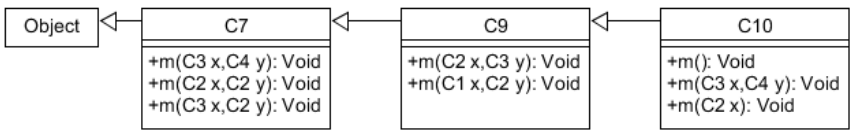


1. Compile time

The type info of super: C9
The type info of the parameter (new C4(), new C4()): (C4, C4)
The scope: C9 and all of its super
The candidates: m(C3,C4), m(C2,C2), m(C3,C2), m(C1,C2), m(C2,C3)
The method signature given: m(C3,C4) because it is the minimum element.

2. Runtime

The type info of super: C9 (the starting place)
The method invoked: m(C3 x, C4 y) in C7



this.m(new C3());

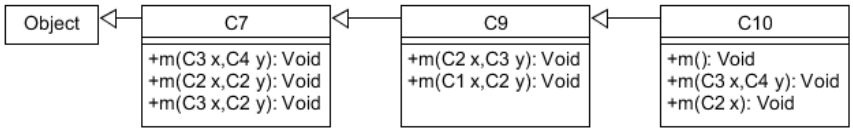
1. Compile time

The type info of this: C10
The type info of the parameter (new C3()): (C3)
The scope: C10 and all of its super
The candidates: m(C2)
The method signature given: m(C2) because it is the minimum element.

m(C2)

2. Runtime

The type info of this: C10 (the starting place)
The method invoked: m(C2 x) in C10




```
super.m(new C3());
```

1. Compile time

The type info of super: C9
The type info of the parameter (new C3()): (C3)
The scope: C9 and all of its super
The candidates: none
The method signature given: none because there is no minimum element.
Therefore, compiler error occurs.

2. Runtime

It cannot be run because it cannot be compiled.