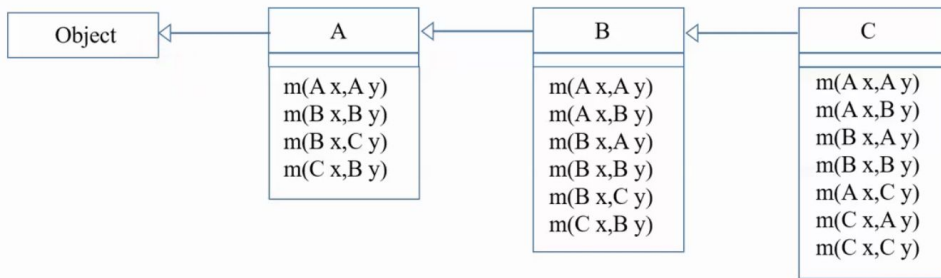


I219: Final Assignment

栗林健太郎 2030006 (2020年12月25日)

Q1

Q1(1)



上図を前提に、以下のコードについてメソッド呼び出しがどのように解決されるかを述べる。

```
(new A()).m(new A(), new A());
```

m(A, A)

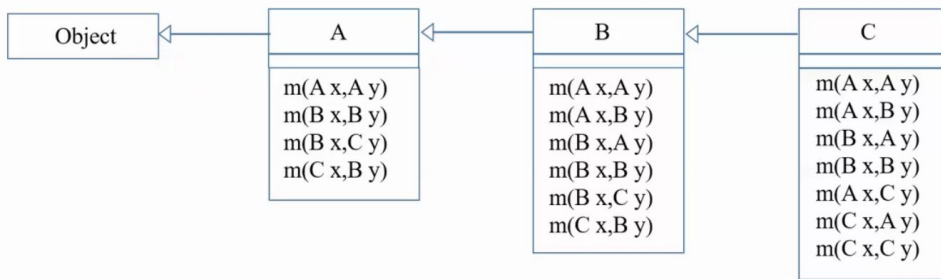
1. コンパイル時

- レシーバの型: A
- 引数の型: A, A
- 探索範囲: Aおよびそのすべての親クラス
- 候補: m(A, A)
- 付与されるメソッドシグネチャ: m(A, A)

2. 実行時

- レシーバの型: A(探索の開始地点)
- 実行されるメソッド: Aのm(A, A)

Q1(2)



上図を前提に、以下のコードについてメソッド呼び出しがどのように解決されるかを述べる。

```
A a = new C(); a.m(a, a);
```

m(A, A)

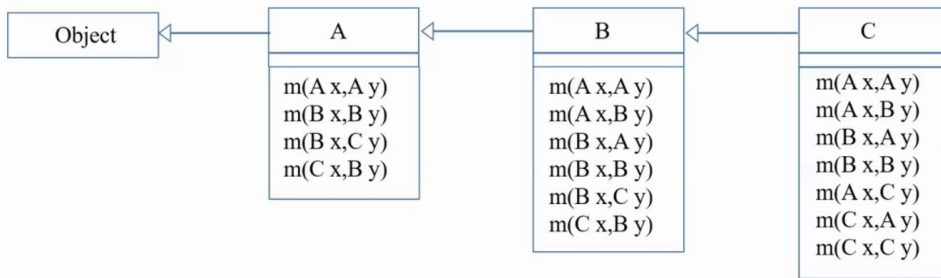
1. コンパイル時

- レシーバの型: A
- 引数の型: A, A
- 探索範囲: Aおよびそのすべての親クラス
- 候補: m(A, A)
- 付与されるメソッドシグネチャ: m(A, A)

2. 実行時

- レシーバの型: C(探索の開始地点)
- 実行されるメソッド: Cのm(A, A)

Q1(3)



上図を前提に、以下のコードについてメソッド呼び出しがどのように解決されるかを述べる。

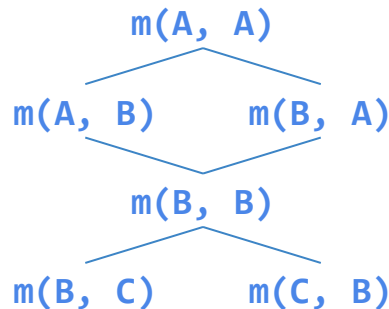
```
B b = new C(); C c = new A(); b.m(c, c);
```

1. コンパイル時

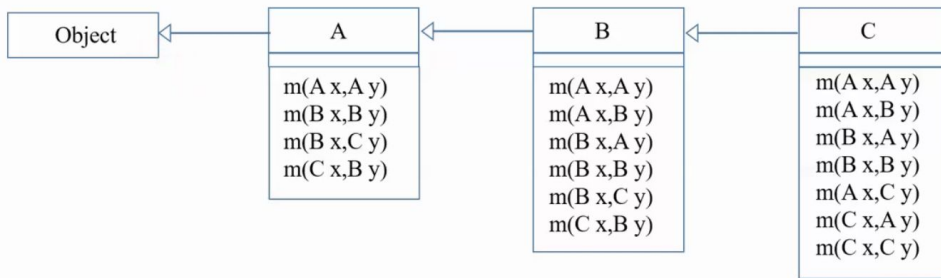
- レシーバの型: B
- 引数の型: C, C
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: なし

2. 実行時

- コンパイルエラーとなるため実行不可能である。



Q1(4)



上図を前提に、以下のコードについてメソッド呼び出しがどのように解決されるかを述べる。

```
C c = new C(); B b = c; A a = b; a.m(b, c);
```

1. コンパイル時

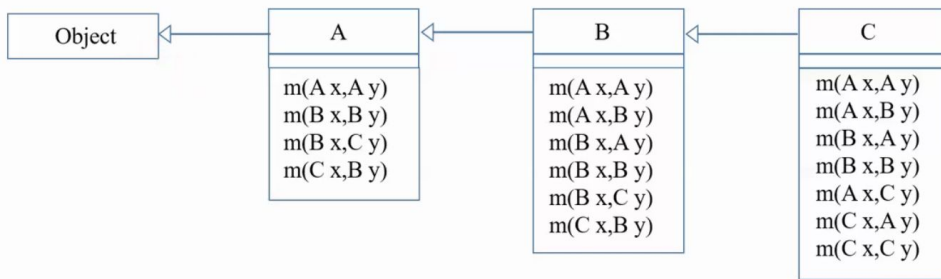
- レシーバの型: A
- 引数の型: B, C
- 探索範囲: Aおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(B, C)

m(A, A)
|
m(B, B)
|
m(B, C)

2. 実行時

- レシーバの型: C(探索の開始地点)
- 実行されるメソッド: Bのm(B, C)

Q1(5)

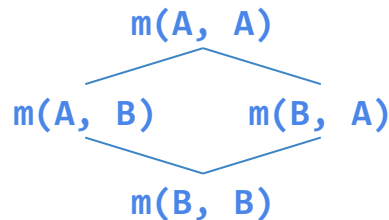


上図を前提に、以下のコードについてメソッド呼び出しがどのように解決されるかを述べる。

```
C c = new C(); B b = c; A a = b; c.m(b, b);
```

1. コンパイル時

- レシーバの型: C
- 引数の型: B, B
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(B, B)

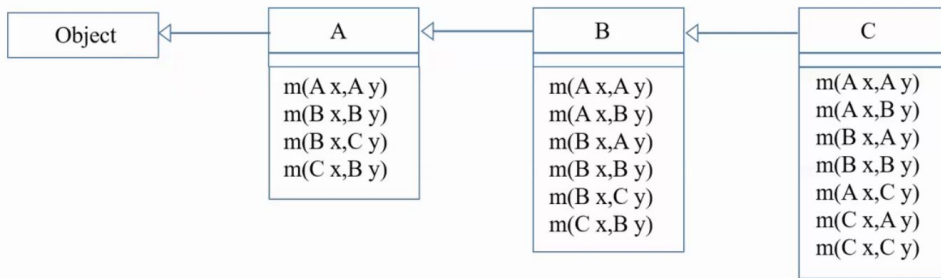


2. 実行時

- レシーバの型: C (探索の開始地点)
- 実行されるメソッド: Cのm(B, B)

Q2

Q2(1)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A x = new C(); B y = new B(); C z = new C(); x.m(z, y);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

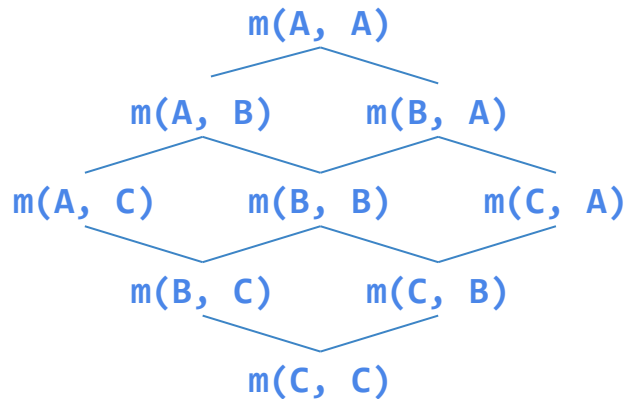
```
x.m(x, x);
```

1. コンパイル時

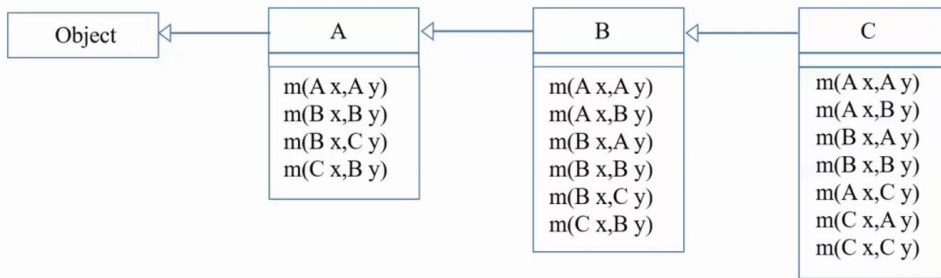
- レシーバの型: C
- 引数の型: C, C
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(C, C)

2. 実行時

- レシーバの型: C(探索の開始地点)
- 実行されるメソッド: Cのm(C, C)



Q2(2)



上図を前提に、以下のメソッド呼び出しが行われる。

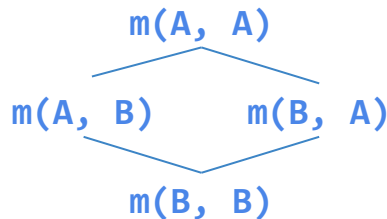
```
A x = new C(); B y = new B(); C z = new C(); x.m(z, y);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(this, this);
```

1. コンパイル時

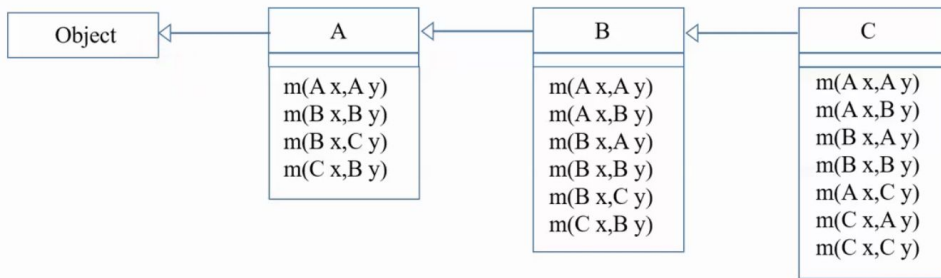
- レシーバの型: B
- 引数の型: B, B
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(B, B)



2. 実行時

- レシーバの型: C(探索の開始地点)
- 実行されるメソッド: Cのm(B, B)

Q2(3)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A x = new C(); B y = new B(); C z = new C(); x.m(z, y);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

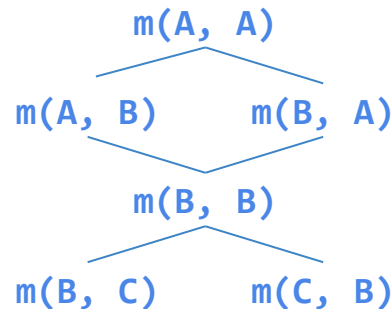
```
this.m(x, x);
```

1. コンパイル時

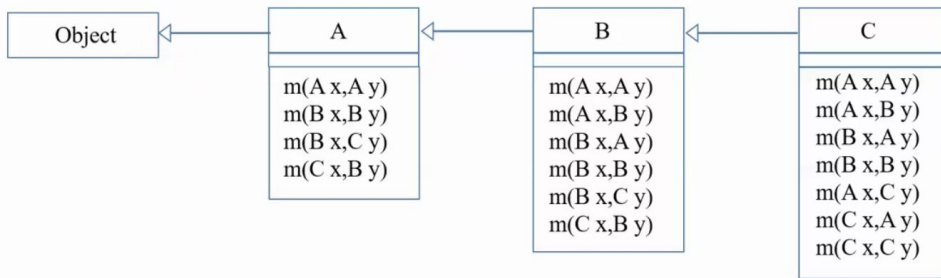
- レシーバの型: B
- 引数の型: C, C
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: なし

2. 実行時

- コンパイルエラーとなるため実行不可能である。



Q2(4)



上図を前提に、以下のメソッド呼び出しが行われる。

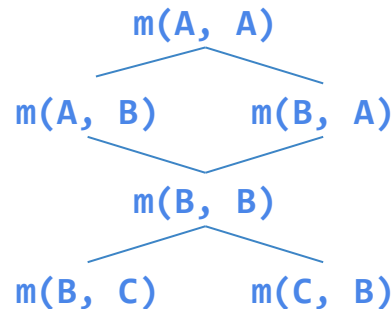
```
A x = new C(); B y = new B(); C z = new C(); x.m(z, y);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
y.m(x, x);
```

1. コンパイル時

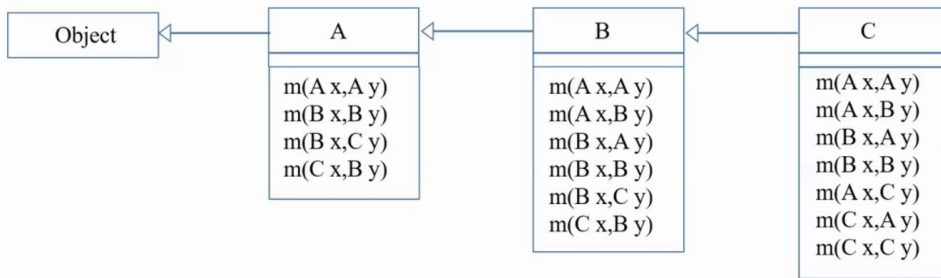
- レシーバの型: B
- 引数の型: C, C
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: なし



2. 実行時

- コンパイルエラーとなるため実行不可能である。

Q2(5)



上図を前提に、以下のメソッド呼び出しが行われる。

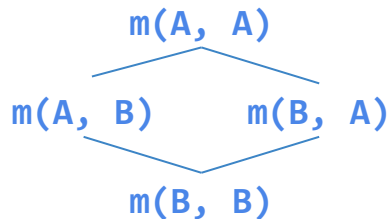
```
A x = new C(); B y = new B(); C z = new C(); x.m(z, y);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
y.m(this, this);
```

1. コンパイル時

- レシーバの型: B
- 引数の型: B, B
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(B, B)

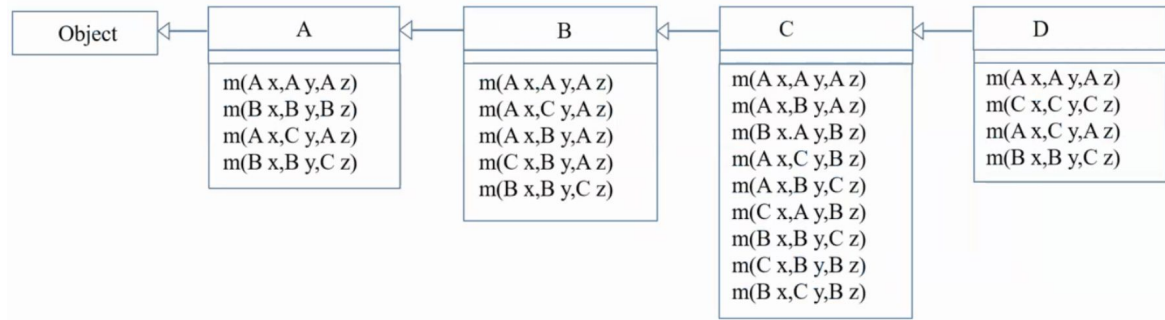


2. 実行時

- レシーバの型: B(探索の開始地点)
- 実行されるメソッド: Bのm(B, B)

Q3

Q3(1)



上図を前提に、以下のメソッド呼び出しが行われる。

```
B a = new B(); B b = new C(); B c = new D(); D d = new D(); d.m(a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(this, this, this);
```

m(A, A, A)

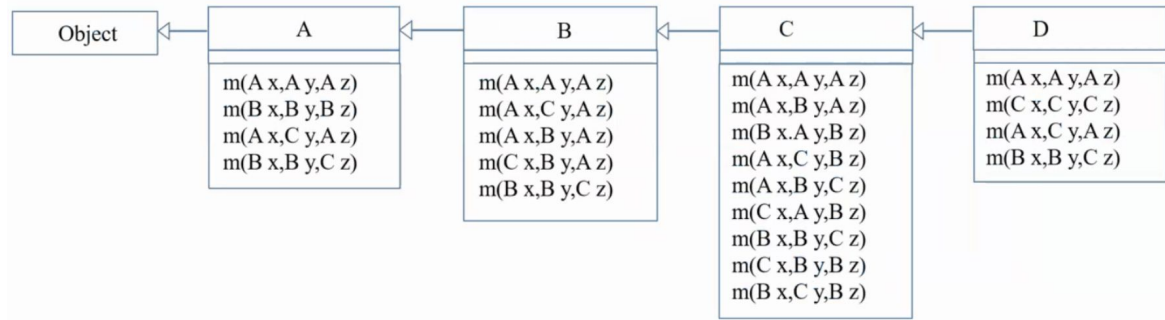
1. コンパイル時

- レシーバの型: A
- 引数の型: A, A, A
- 探索範囲: Aおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(A, A, A)

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Dのm(A, A, A)

Q3(2)



上図を前提に、以下のメソッド呼び出しが行われる。

```
B a = new B(); B b = new C(); B c = new D(); D d = new D(); d.m(a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
(new D()).m(x, this, z);
```

);

1. コンパイル時

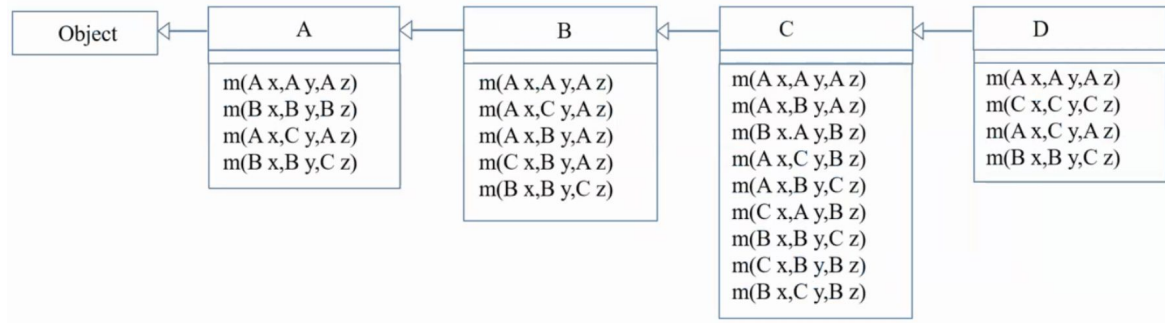
- レシーバの型: D
- 引数の型: B, A, B
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(B, A, B)

```
m(A, A, A)
|
m(B, A, B)
```

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Cのm(B, A, B)

Q3(3)



上図を前提に、以下のメソッド呼び出しが行われる。

```
B a = new B(); B b = new C(); B c = new D(); D d = new D(); d.m(a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
(new C()).m(new D(), new D(), new D());
```

);

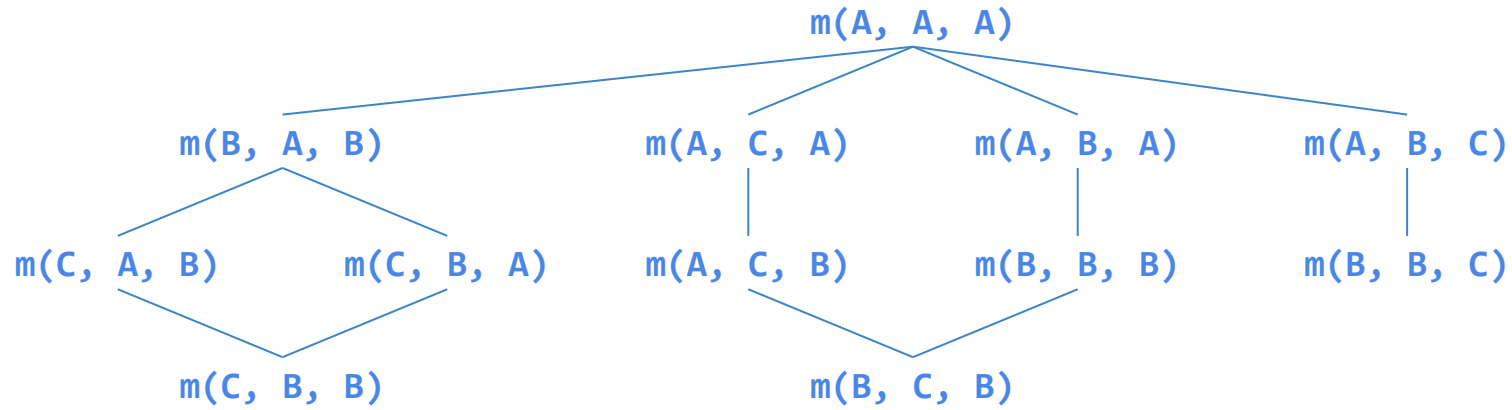
1. コンパイル時

- レシーバの型: C
- 引数の型: D, D, D
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: なし

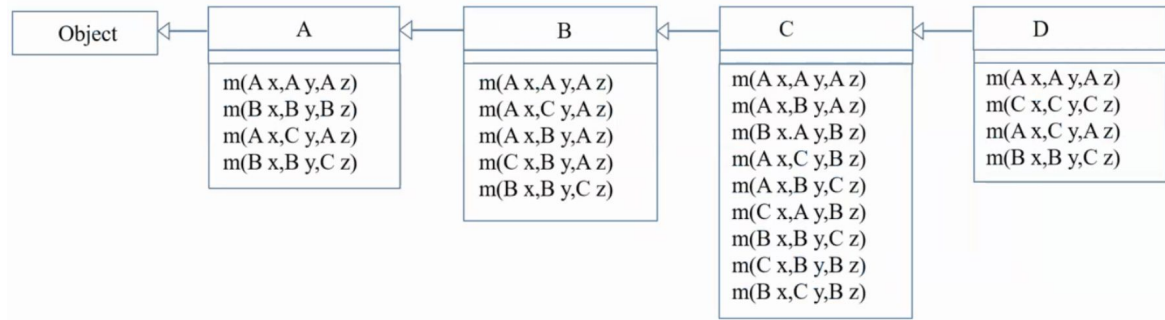
2. 実行時

- コンパイルエラーとなるため実行不可能である。

Q3(3)



Q3(4)



上図を前提に、以下のメソッド呼び出しが行われる。

```
B a = new B(); B b = new C(); B c = new D(); D d = new D(); d.m(a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
(new D()).m(new D(), new D(), new D());
```

);

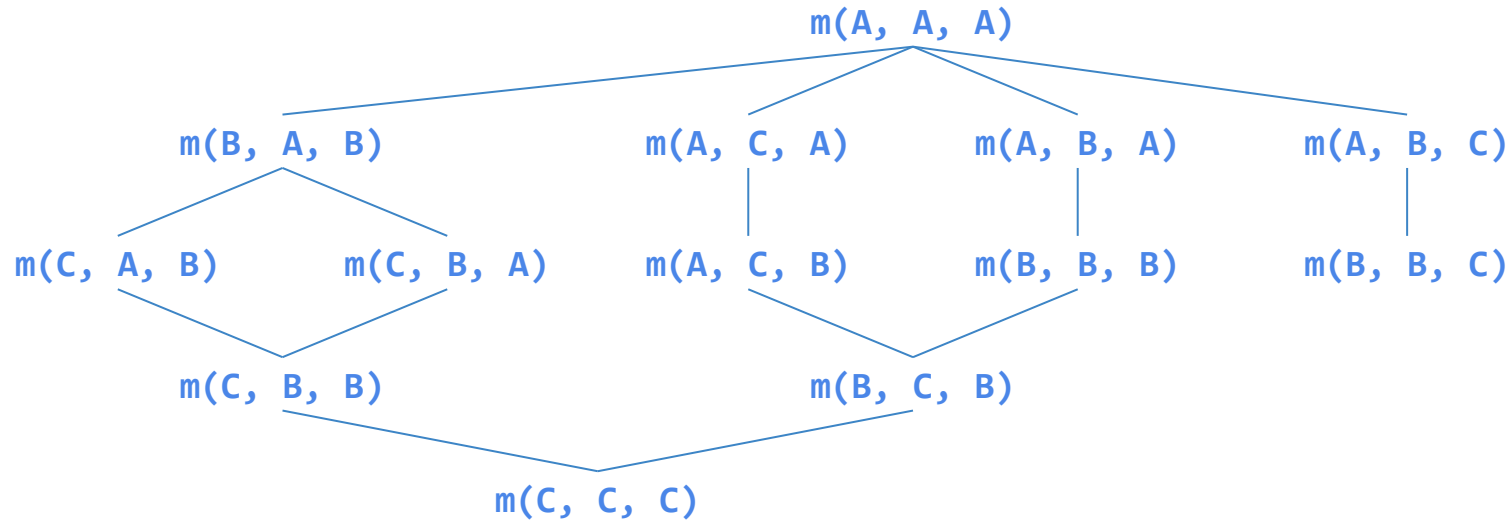
1. コンパイル時

- レシーバの型: D
- 引数の型: D, D, D
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(C, C, C)

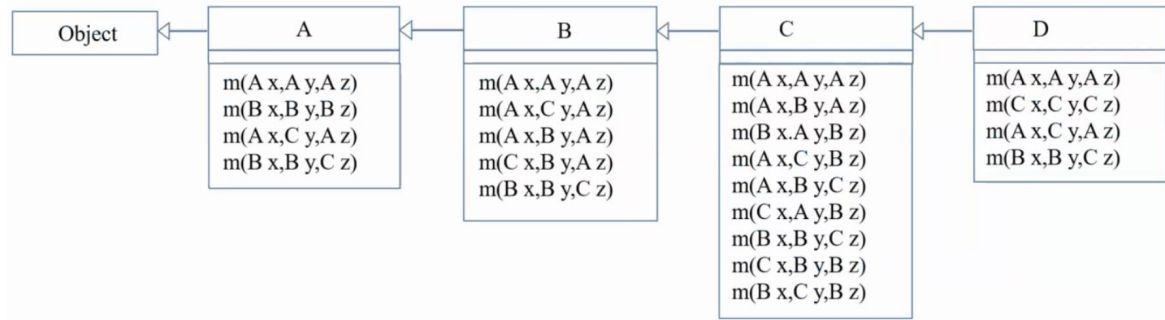
2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Cのm(C, C, C)

Q3(4)



Q3(5)



上図を前提に、以下のメソッド呼び出しが行われる。

```
B a = new B(); B b = new C(); B c = new D(); D d = new D(); d.m(a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
(new C()).m(new D(), x, z);
```

};

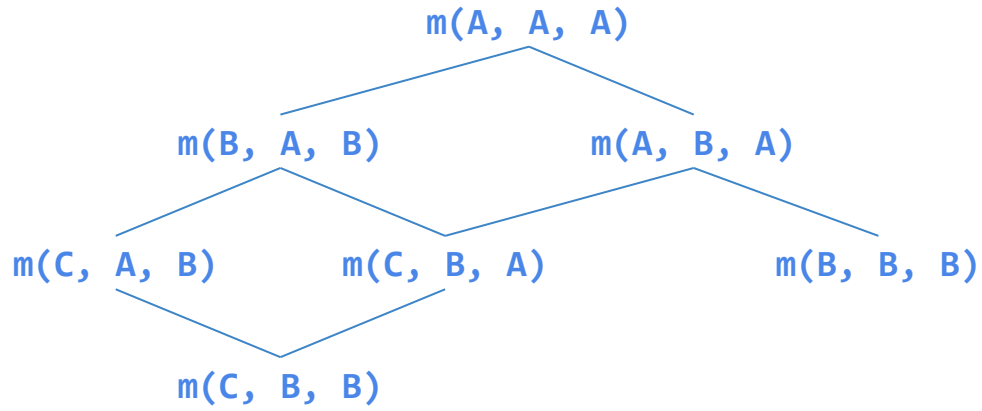
1. コンパイル時

- レシーバの型: C
- 引数の型: D, B, B
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(C, B, B)

2. 実行時

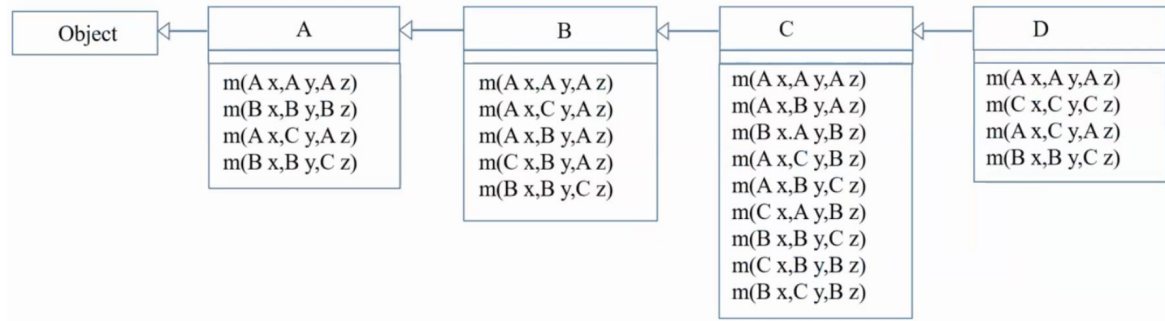
- レシーバの型: C(探索の開始地点)
- 実行されるメソッド: Cのm(C, B, B)

Q3(5)



Q4

Q4(1)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); B b = new C(); C c = new D(); D d = new D(); d.m(c, b, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(this, this, this);
```

1. コンパイル時

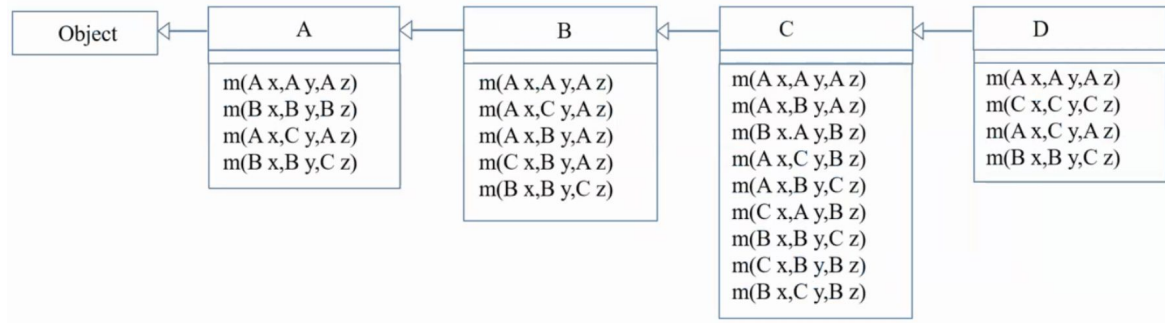
- レシーバの型: B
- 引数の型: B, B, B
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(B, B, B)

```
m(A, A, A)
|
m(B, A, B)
```

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Aのm(B, B, B)

Q4(2)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); B b = new C(); C c = new D(); D d = new D(); d.m(c, b, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
x.m(x, x, x);
```

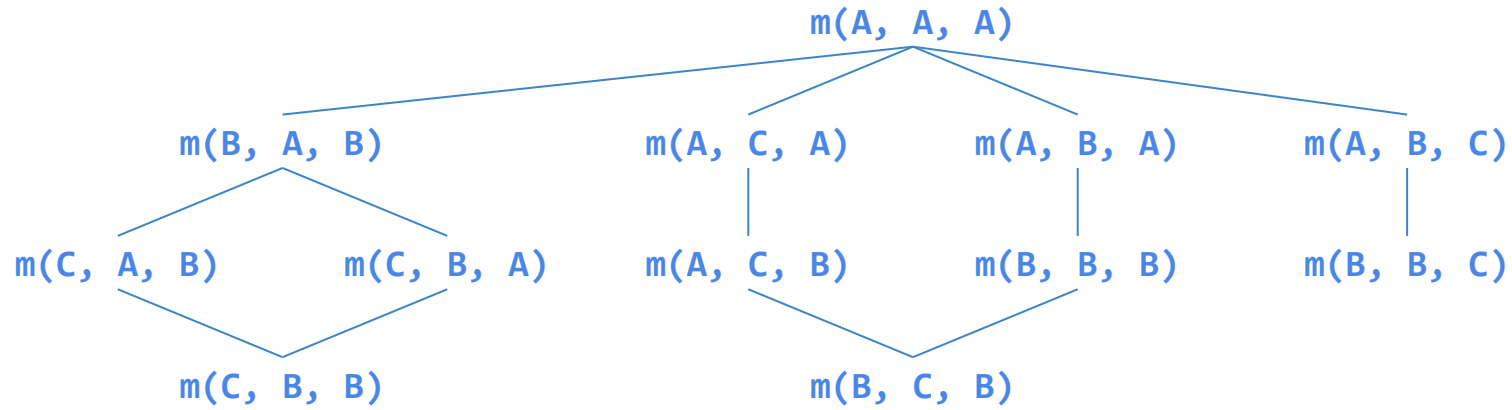
1. コンパイル時

- レシーバの型: C
- 引数の型: C, C, C
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: なし

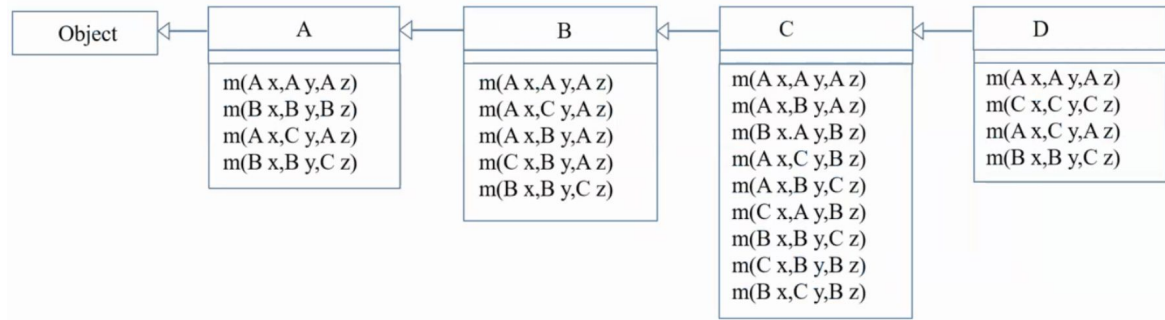
2. 実行時

- コンパイルエラーとなるため実行不可能である。

Q4(2)



Q4(3)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); B b = new C(); C c = new D(); D d = new D(); d.m(c, b, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
(new D()).m(x, x, x);
```

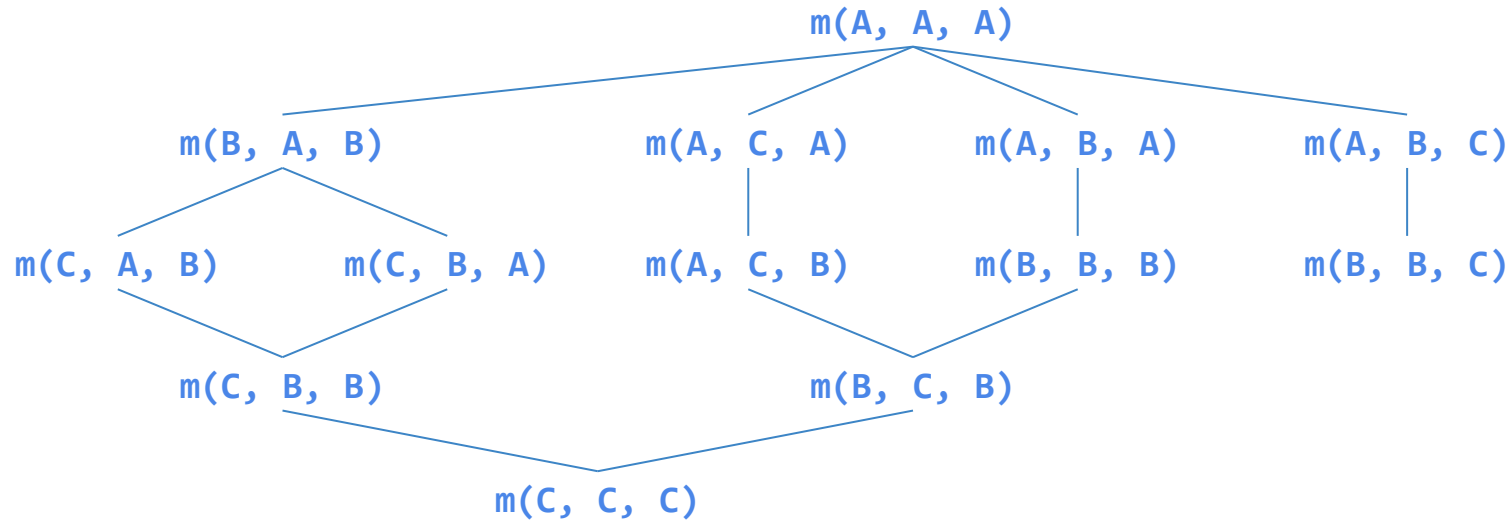
1. コンパイル時

- レシーバの型: D
- 引数の型: C, C, C
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(C, C, C)

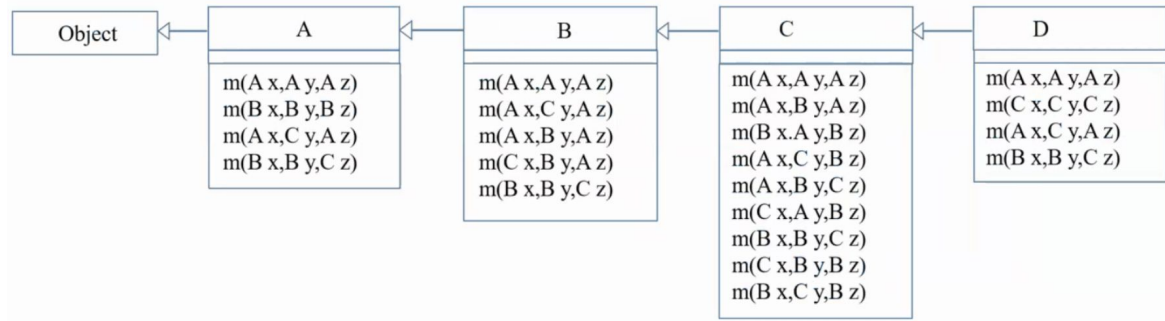
2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Dのm(C, C, C)

Q4(3)



Q4(4)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); B b = new C(); C c = new D(); D d = new D(); d.m(c, b, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
x.m(z, this, z);
```

1. コンパイル時

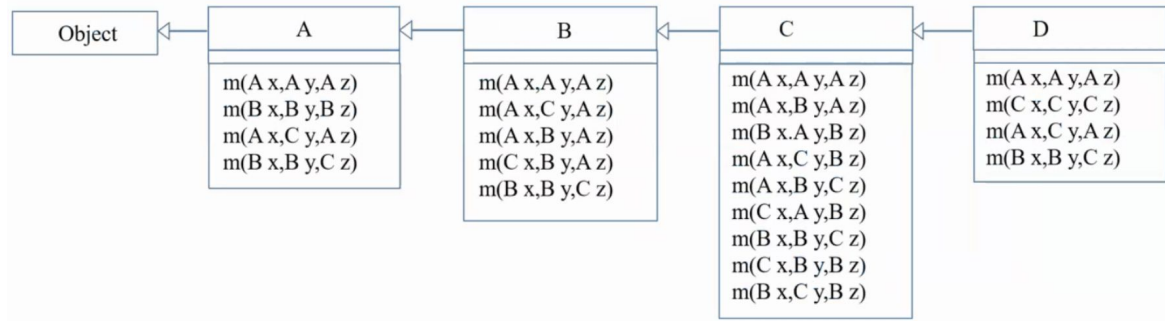
- レシーバの型: C
- 引数の型: A, B, A
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(A, B, A)

```
m(A, A, A)
|
m(A, B, A)
```

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Cのm(A, B, A)

Q4(5)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); B b = new C(); C c = new D(); D d = new D(); d.m(c, b, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(z, this, z);
```

1. コンパイル時

- レシーバの型: B
- 引数の型: A, B, A
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(A, B, A)

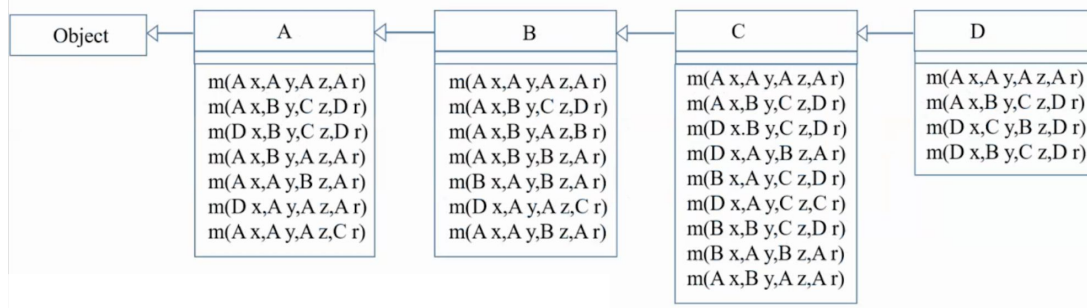
m(A, A, A)
|
m(A, B, A)

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Cのm(A, B, A)

Q5

Q5(1)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); A b = new C(); A c = new D(); D d = new D(); d.m(d, a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(this, this, this, this);
```

m(A, A, A, A)

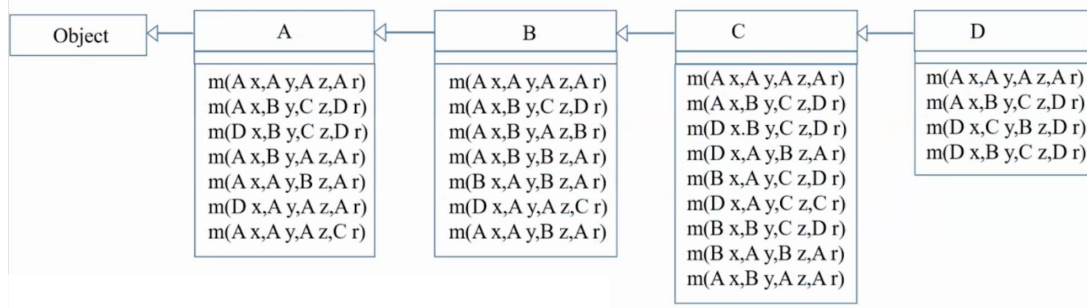
1. コンパイル時

- レシーバの型: A
- 引数の型: A, A, A
- 探索範囲: Aおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(A, A, A, A)

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Dのm(A, A, A, A)

Q5(2)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); A b = new C(); A c = new D(); D d = new D(); d.m(d, a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(new D(), new B(), new D(), new D());
```

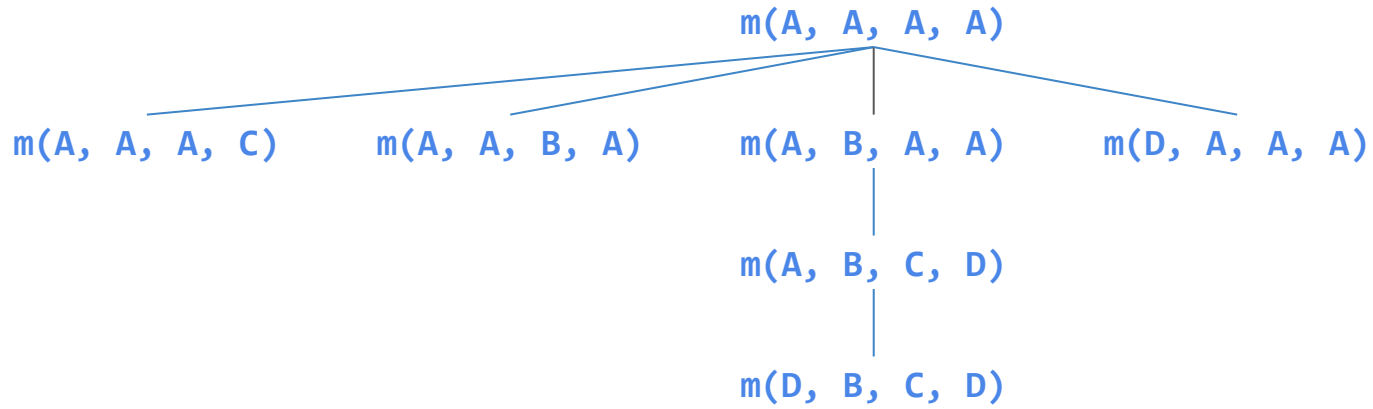
1. コンパイル時

- レシーバの型: A
- 引数の型: D, B, D, D
- 探索範囲: Aおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(D, B, C, D)

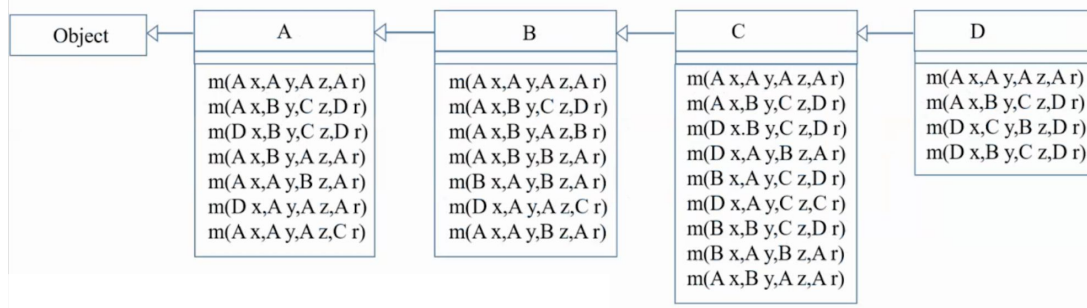
2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Dのm(D, B, C, D)

Q5(2)



Q5(3)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); A b = new C(); A c = new D(); D d = new D(); d.m(d, a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
r.m(new A(), new B(), x, x);
```

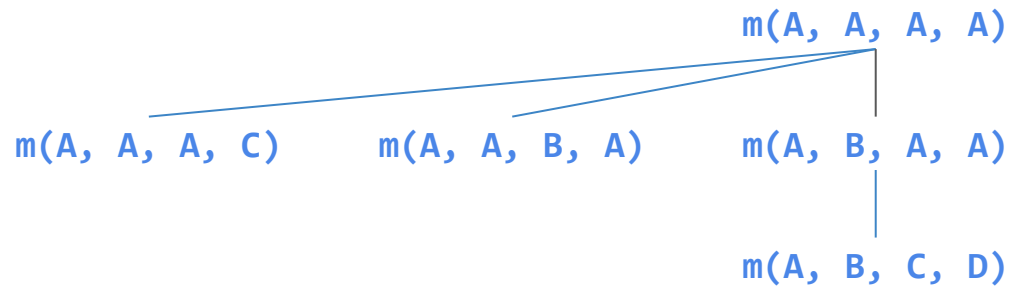
1. コンパイル時

- レシーバの型: A
- 引数の型: A, B, D, D
- 探索範囲: Aおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(A, B, C, D)

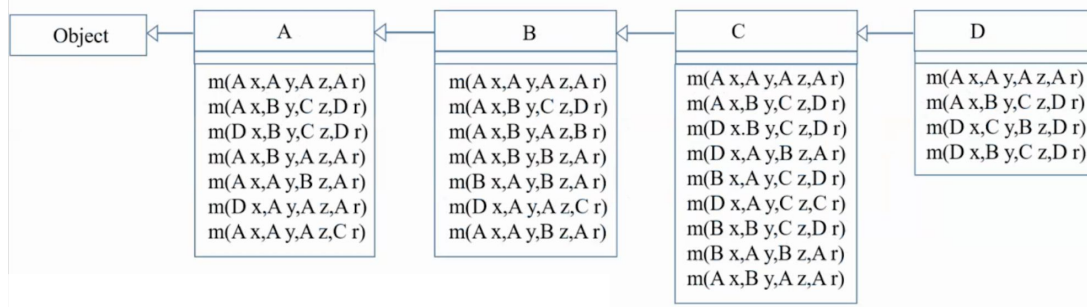
2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Aのm(A, B, C, D)

Q5(3)



Q5(4)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); A b = new C(); A c = new D(); D d = new D(); d.m(d, a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
x.m(x, x, x, x);
```

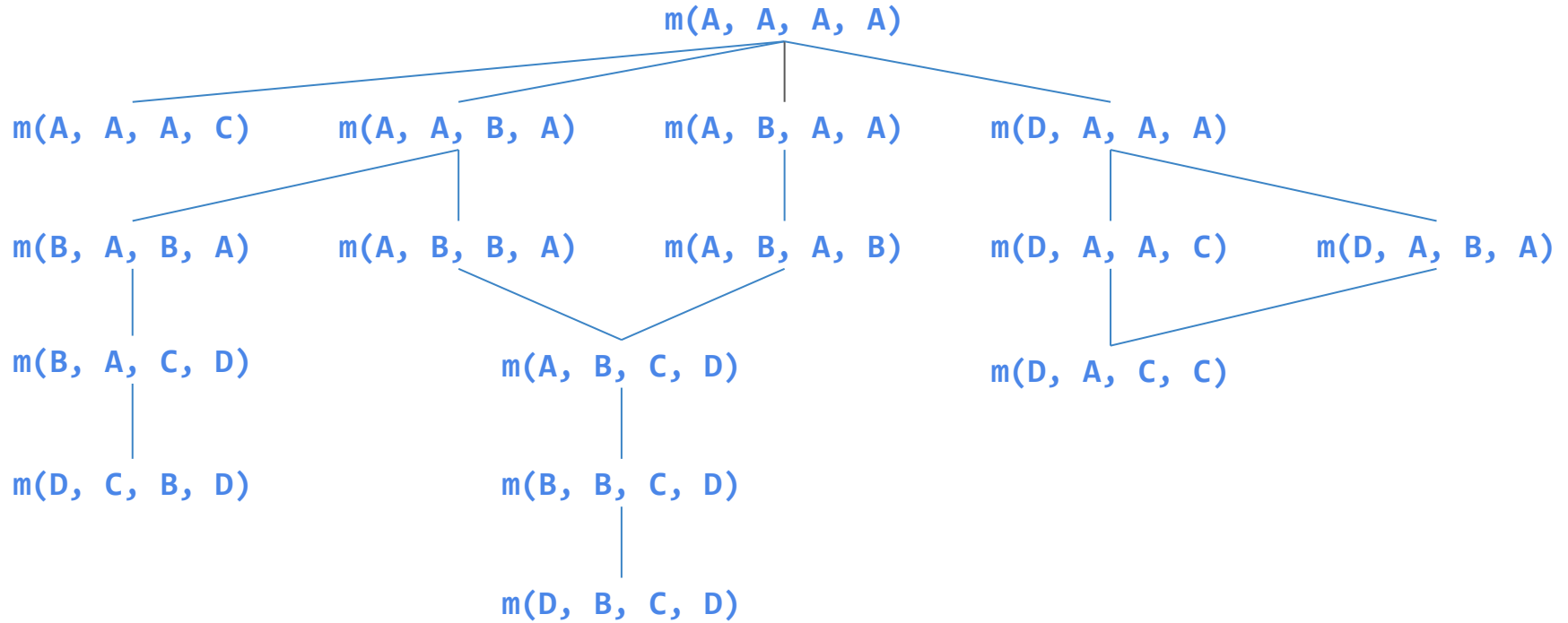
1. コンパイル時

- レシーバの型: D
- 引数の型: D, D, D, D
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: なし

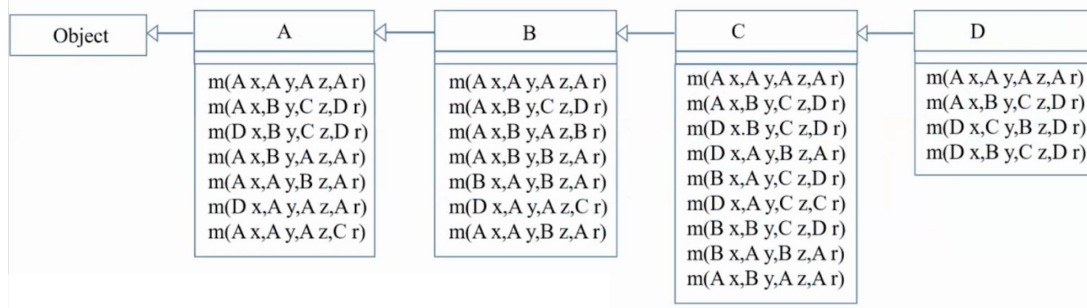
2. 実行時

- コンパイルエラーとなるため実行不可能である。

Q5(4)



Q5(5)



上図を前提に、以下のメソッド呼び出しが行われる。

```
A a = new B(); A b = new C(); A c = new D(); D d = new D(); d.m(d, a, b, c);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
x.m(x, new C(), new B(), x);
```

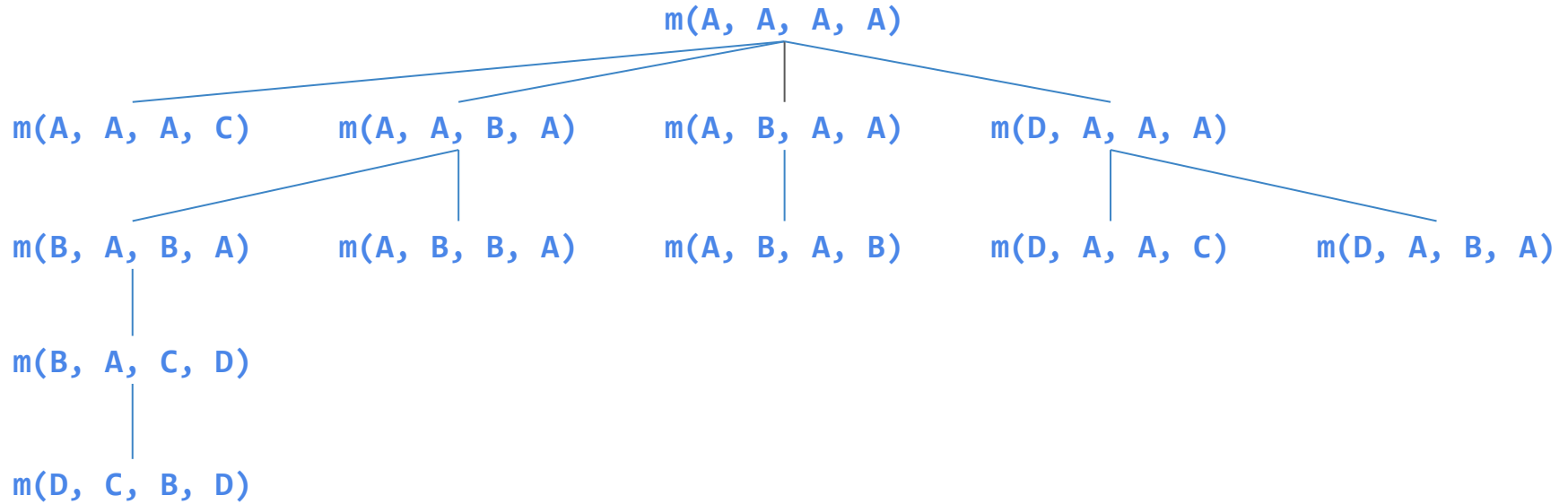
1. コンパイル時

- レシーバの型: D
- 引数の型: D, C, B, D
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(D, C, B, D)

2. 実行時

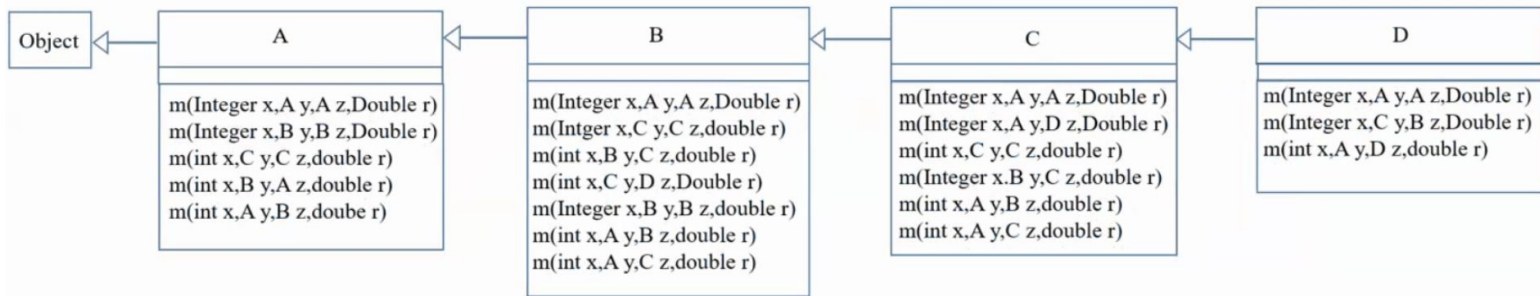
- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Dのm(D, C, B, D)

Q5(5)



Q6

Q6(1)



上図を前提に、以下のメソッド呼び出しが行われる。

```
Double a = new Double(3.14); C c = new D(); D d = new D(); d.m(2, c, d, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
z.m(2, y, y, 3.14);
```

1. コンパイル時

- レシーバの型: D
- 引数の型: int, C, C, double
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(int, C, C, double)

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Cのm(int, C, C, double)

Q6(1)

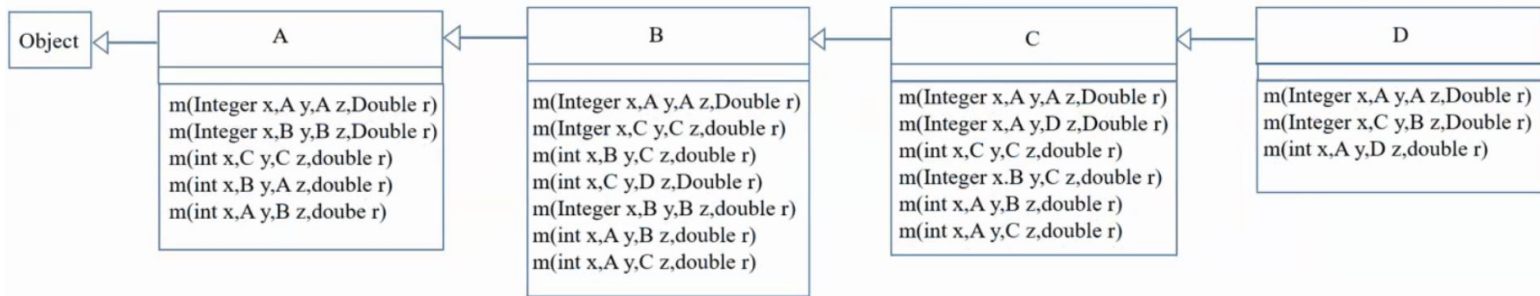
m(Integer, B, B, double)
|
m(Integer, B, C, double)
|
m(Integer, C, C, double)

m(int, A, B, double)
|
m(int, A, C, double)
|
m(int, B, C, double)
|
m(int, C, C, double)

m(int, B, A, double)

m(Integer, A, A, Double)
|
m(Integer, B, B, Double)
|
m(Integer, C, B, Double)

Q6(2)



上図を前提に、以下のメソッド呼び出しが行われる。

```
Double a = new Double(3.14); C c = new D(); D d = new D(); d.m(2, c, d, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
z.m(2, new B(), new B(), r);
```

1. コンパイル時

- レシーバの型: D
- 引数の型: int, B, B, Double
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: なし

2. 実行時

- コンパイルエラーとなるため実行不可能である。

Q6(2)

`m(Integer, B, B, double)`

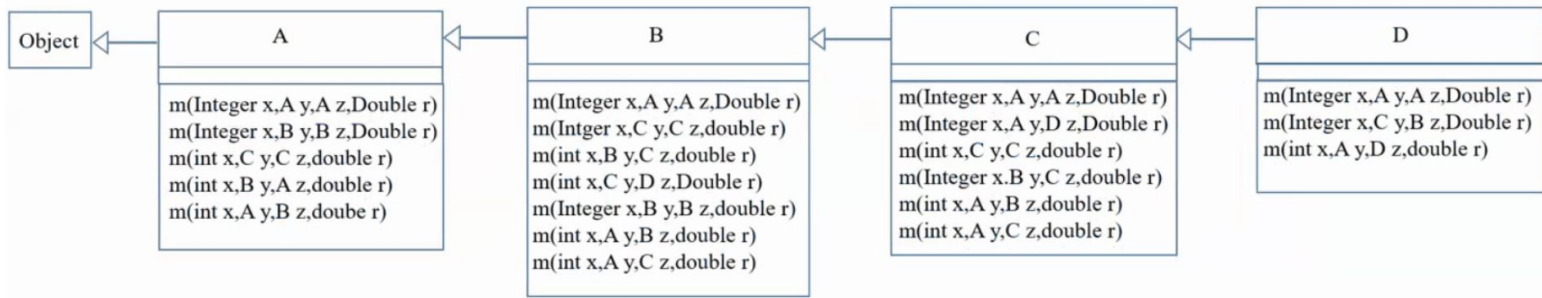
`m(int, A, B, double)`

`m(int, B, A, double)`

`m(Integer, A, A, Double)`

|
`m(Integer, B, B, Double)`

Q6(3)



上図を前提に、以下のメソッド呼び出しが行われる。

```
Double a = new Double(3.14); C c = new D(); D d = new D(); d.m(2, c, d, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
this.m(2, new A(), new A(), 3.14);
```

1. コンパイル時

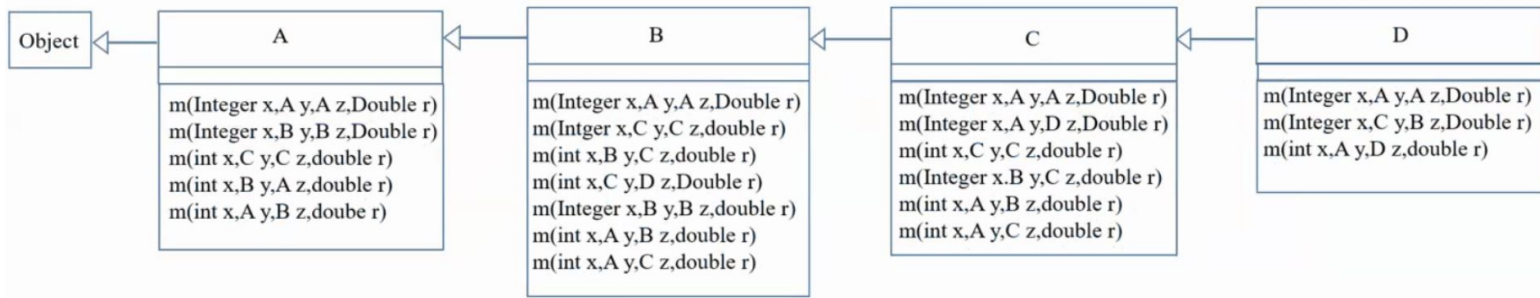
- レシーバの型: B
- 引数の型: int, A, A, double
- 探索範囲: Bおよびそのすべての親クラス
- 候補: 右図の通り
- 付与されるメソッドシグネチャ: m(Integer, A, A, Double)

m(Integer, A, A, Double)

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Dのm(Integer, A, A, Double)

Q6(4)



上図を前提に、以下のメソッド呼び出しが行われる。

```
Double a = new Double(3.14); C c = new D(); D d = new D(); d.m(2, c, d, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
z.m(new Integer(2), new B(), new B(), r);
```

1. コンパイル時

- レシーバの型: D
- 引数の型: Integer, B, B, Double
- 探索範囲: Dおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: m(Integer, B, B, Double)

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Aのm(Integer, B, B, Double)

Q6(4)

`m(Integer, B, B, double)`

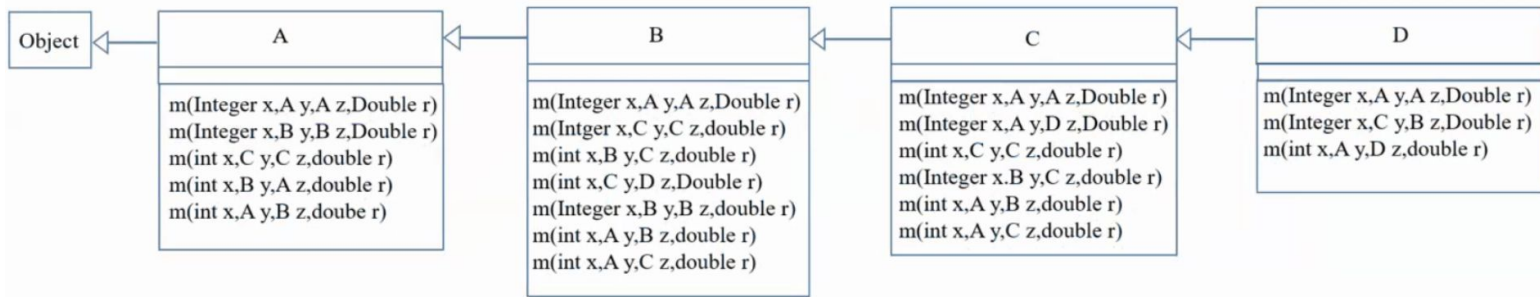
`m(int, A, B, double)`

`m(int, B, A, double)`

`m(Integer, A, A, Double)`

|
`m(Integer, B, B, Double)`

Q6(5)



上図を前提に、以下のメソッド呼び出しが行われる。

```
Double a = new Double(3.14); C c = new D(); D d = new D(); d.m(2, c, d, a);
```

その際、呼び出されたメソッドが以下のメソッド呼び出しを含む場合について述べる。

```
y.m(new Integer(2), new A(), z, r);
```

1. コンパイル時

- レシーバの型: C
- 引数の型: Integer, A, D, Double
- 探索範囲: Cおよびそのすべての親クラス
- 候補: 次ページ図の通り
- 付与されるメソッドシグネチャ: `m(Integer, A, D, Double)`

2. 実行時

- レシーバの型: D(探索の開始地点)
- 実行されるメソッド: Cの`m(Integer, A, D, Double)`

Q6(5)

m(int, A, B, double)

|

m(int, A, C, double)

m(Integer, A, A, Double)

|

m(Integer, A, D, Double)

m(int, A, D, double)