

Esercizio: overloading e gerarchia

Considerare il seguente codice java:

```
public class D { };

public class C extends D {
    void m(Object o, long x, long y) {...} // C1
    void m(String s, int x, long y) {...} // C2
    void m(Object o, int x, long y) {...} // C3
    void m(String s, long x, int y) {...} // C4
}

public class B extends C {
    void m(Object o, long x, long y) {...} // B1
    void m(String s, int x, long y) {...} // B2
    void m(Object o, int x, long y) {...} // B3
    void m(String s, long x, int y) {...} // B4
}
...

public void Main (String [] args){
    Object theObject;
    String theString;
    int theInt;
    long theLong;

    C aC = new C();
    D aD = new D();
    D anotherD = new C();
    B aB = new B();
}
```

Quali delle seguenti invocazioni dentro il body di Main sono corrette, e quelle corrette quale definizione invocherebbero?

1. Deciderlo in via analitica
2. Implementarlo e provarlo.

```
aC.m(theString, theInt, theLong)
aC.m(theString, theInt, theInt)
aC.m(theString, theLong, theInt)
aC.m(theString, theLong, theLong)
aC.m(theObject, theLong, theLong)
aC.m(theObject, theInt, theInt)
```

```
aD.m(theString, theInt, theLong)
aD.m(theString, theInt, theInt)
aD.m(theString, theLong, theInt)
aD.m(theString, theLong, theLong)
aD.m(theObject, theLong, theLong)
aD.m(theObject, theInt, theInt)
```

```
anotherD.m(theString, theInt, theLong)
anotherD.m(theString, theInt, theInt)
anotherD.m(theString, theLong, theInt)
anotherD.m(theString, theLong, theLong)
anotherD.m(theObject, theLong, theLong)
anotherD.m(theObject, theInt, theInt)
```

```
aB.m(theString, theInt, theLong)
aB.m(theString, theInt, theInt)
aB.m(theString, theLong, theInt)
aB.m(theString, theLong, theLong)
aB.m(theObject, theLong, theLong)
aB.m(theObject, theInt, theInt)
```