

1: ((Bird)dodo).ability : Fly

Dodo ist eine Subclass von Bird. Das Attribut wird zur Compilezeit ausgewählt und daher wird das Attribut *ability* von Bird gewählt

2: dodo.ability : Run

Hier ist dodo zur Compilezeit auch vom Typen Dodo, daher wird hier auch das Attribut *ability* aus der Dodo Class gewählt

3: dodo.getAbility() : Fly

In getAbility aus der Dodo Class steht super.getAbility(), daher wird die Methode aus der Superclass aufgerufen und somit das Attribut aus der Superclass Bird zurück gegeben

4: parrot.allAbilities() : Fly Talk

In allAbilities aus der Parrot Class steht super.allAbilities() + " " + ability. Das *super.allAbilities()* gibt dabei die Abilities aus der Superclass zurück, welches hier das Fly ist und *ability* gibt hier das Attribut ability aus der Parrot Class zurück, also Talk

5: parrot.ability : Talk

Da parrot zur Compiletime vom Typen Parrot ist, wird hier das Attribut *ability* von der Class Parrot zurück gegeben

6: carsten.ability : Fly

carsten ist zwar vom Typen Dodo, aber zur Compiletime vom Typen Bird, daher wird hier das Attribut aus der Bird Class zurück gegeben

7: ((Bird)carsten).allAbilities() : Run

Methoden werden zur Runtime ausgewertet, daher wird hier die Methode auf dem Objekt carsten angewandt, welches eigentlich vom Typen Dodo ist und somit dann auch die Methode aus der Dodo Class aufgerufen wird

8: einstein.allAbilities() : Fly Talk

Methoden werden zur Runtime ausgeführt, daher wird die Methode aus dem Parrot aufgerufen

9: einstein.getAbility() : Fly

Da in Parrot die Methode getAbility nicht überschrieben wird, wird die Methode aus der Bird Class verwendet

10: ((Parrot)einstein).ability : Talk

Da das Attribut zur Compiletime bestimmt wird, wird hier das Attribut aus Parrot verwendet

Parrot ist keine Subclass von Dodo und umgekehrt gilt das gleiche. Zwei unabhängige Objekte können nicht einfach frei gecasted werden