

# OOP Aufgabenblatt 7

Donnerstag, 23. Juni 2022 18:26

1a)

Aufruf	Methode	Begründung
add(1.0, 2.0);	----	Keine passende Signatur. Für Methode 2 müsste z.B. das double "2.0" explizit zu Long konvertiert werden.
add(1, 2);	Method 1, int add(long a, int b)	1 wird implizit zu long konvertiert.
add(2, 1L);	Method 2, int add(double a, float b)	2 wird implizit zu double und 1L zu float konvertiert.
print(c, c);	Method 3, void print(B b, C c)	Das erste Argument wird implizit zu Typ B konvertiert. Das ist möglich, da C von B vererbt ist.
print(a, c);	Method 4, void print(A a, B b)	Method4 und nicht Method3, weil A kann nicht implizit zu B konvertiert werden.
print(c, b);	Method 4, void print(A a, B b)	C ist gleichzeitig A, B ist B (dank Polymorphie).
print(b, a);	----	Keine passende Signatur. Das 2te Argument ist sozusagen "zu groß" für alle definierten Methoden.

1b)

Aufruf	Ausgabe	Begründung
System.out.println(p1.x);	1	Statisches Binden - Java erlaubt Polymorphism mit Fields nicht.
p1.print(c);	4	Dynamisches Binden - überschreibende print-Method von Klasse Print2 mit derselben Signatur (i.e. print(B b) ) wird aufgerufen, und die Variable x ist im Gültigkeitsbereich von Print2 überdeckt.
((Print2)p1).print(c);	5	print-Method von Klasse Print2 mit der passenden Signatur print(C c) wird aufgerufen, und int Feld x ist in der Klasse Print2 mit x=3 überdeckt.
((Print1)p2).print(b);	4	Dynamisches Binden - Print2::print(B b) wird aufgerufen.
((Print1)p2).print(c);	4	Dynamisches Binden - Print2::print(B b) wird aufgerufen (Signatur wird vor dem dynamischen Binden festgelegt, daher nicht Print2::print(C c) )
p2.print(c);	5	Selbsterklärend.
((Print2)p1).print(a);	Fehlermeldung.	(p1 instanceof Print2) ist False, daher "downcasting" (Typumwandlung zur oberen Klasse) mittels cast-Operator ist unmöglich.