

$$\rightarrow b.x = 2.5 f$$

$$b.y = 1$$

$$\Rightarrow \text{System.out.print}(A.x) = 13$$

$$\text{System.out.print}(b.x) = 2.5 f$$

$$\text{System.out.print}((A) b.y) = -15$$

$$\text{System.out.print}(b.y) = 1$$

4) Kein Konstruktor mit passender Signatur vorhanden (int) in B \rightarrow Type cast zu float:

\Rightarrow Aufruf des float Konstruktors in B
 \rightarrow Impliziter float Konstruktoraufruf in A mit 3.0 f
 \rightarrow kein passender Konstruktor \Rightarrow Type cast zu 3.0 von Typ double
 $\rightarrow A.y = y + x = -2$

$$\text{super.y}++ \Rightarrow A.y = -15 + 1 = -14$$

$$\Rightarrow \text{System.out.print}(ab.y) = -1$$

$$\text{System.out.print}((B) ab.y) = 1$$

5b) 1. Parameter int i; A an

Aufruf Methode in A passender Signatur $\Rightarrow A.f(1A)$

2. Parameter long; A ab

Aufruf Methode 1. f(1A) da keine passende Methode in A vorhanden, und Typecasting vor Autoboxing \Rightarrow long zu double konvertiert $\Rightarrow A.f(1A)$ wird ausgeführt

3. Parameter $\text{int } i$, B ab

Passende Signatur Methode in $B \Rightarrow B.f(1B)$ wird ausgeführt

4. Parameter $\text{double } d$, A ab

Kein passende Methode verfügbar in $B \Rightarrow B$ erbt Methoden \Rightarrow Typecast von B Objekt zu A Objekt $\Rightarrow A.f(1A)$ wird ausgeführt

5. Parameter $\text{int } i$, A ab

Eigentlich passende Signatur $A.f(1A)$ in Klasse A .
Da ab zu Laufzeit jedoch B Objekt wird, wird die Methode $A.f(1A)$ von $B.f(1A)$ überschrieben

6. Parameter $\text{int } i$, B b

$B.f(1B)$ wird ausgeführt, da ab zu Laufzeit B Objekt und A keine passende Methode besitzt.