

## Aufgabe 5a)

a) 1) Expliziter Aufruf von `public A(int x)` mit 5

→ Implizit `super()` aus Object Aufruf

→ Setze a1.x auf  $x+2 = 5+2 = 7$

→ Setze a1.y auf  $y-x = 7-7 = 0$

⇒ (globale Variable aus A; a1.x wird aufgerufen)

→ A.x und A.y bleiben unverändert da nicht explizit mit `A.x = ...` aufgerufen.

⇒ `System.out.print(A.x) = 0`

⇒ `System.out.print(a1.y) = 0`

a) 2) Expliziter Aufruf von `public A()`

→ Implizit `this(x)` also `public A(int x)` mit  $x=0$  aufgerufen

→ Implizit `super()` aus Object wird aufgerufen

⇒  $a2.x = x+2 = 0+2 = 2$

⇒  $a2.y = y-x = 7-2 = 5$

→ Anschließend wird A.x auf  $x+1 = 0+1 = 1$  gesetzt (da `this.` fehlt und A.x größere Sichtbarkeit besitzt als `a1.x`)

⇒ `System.out.print(A.x) = 1`

⇒ `System.out.print(a2.y) = 5`

a)3) Expliziter Aufruf von `public B()`  
→ impliziter Aufruf von `public A()`  
→ `this(x)` also `public A()` wird implizit aufgerufen  
→ impliziter Aufruf von `super()` in Objekt

$$\Rightarrow ((A) b1).x = x + 2 = 1 + 2 = 3$$

$$\Rightarrow ((A) b1).y = y - x = 7 - 3 = 4$$

`A.x` wird um 1 erhöht  $\Rightarrow A.x = 1 + 1 = 2$

$\Rightarrow B.x$  wird auf 2.5f gesetzt ( $1.5f + 1$ )

$$\Rightarrow \text{System.out.print}(A.x) = 2$$

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

$$\Rightarrow \text{System.out.print}((A) b).y = 4$$

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

a)4) `public B(float x)` wird <sup>implizit</sup> aufgerufen und übergebene Variable `x` wird zu `3.0f` gecasted (da kein int Konstruktor in `B`)

$\Rightarrow$  `super(x)` ruft `public B(3.0f)` auf (implizit)  
→ `3.0f` wird zu `3.0` gecasted da kein Konstrukt für float in `A`.

$\Rightarrow$  `super.y++` setzt `((A) ab).y` auf 8

$\Rightarrow$  implizit Aufruf von `public A(double x)`  
mit `3.0`  $\Rightarrow ((A) ab).y = 8 + 3.0 = 11$  (da `((A) ab).y` Typ int hat).

Da `B.y` nicht statisch ist bleibt es 1

$$\Rightarrow \text{System.out.print}(ab.y) = 11; \text{System.out.print}((B) ab).y = 1$$

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.