

Programmierung

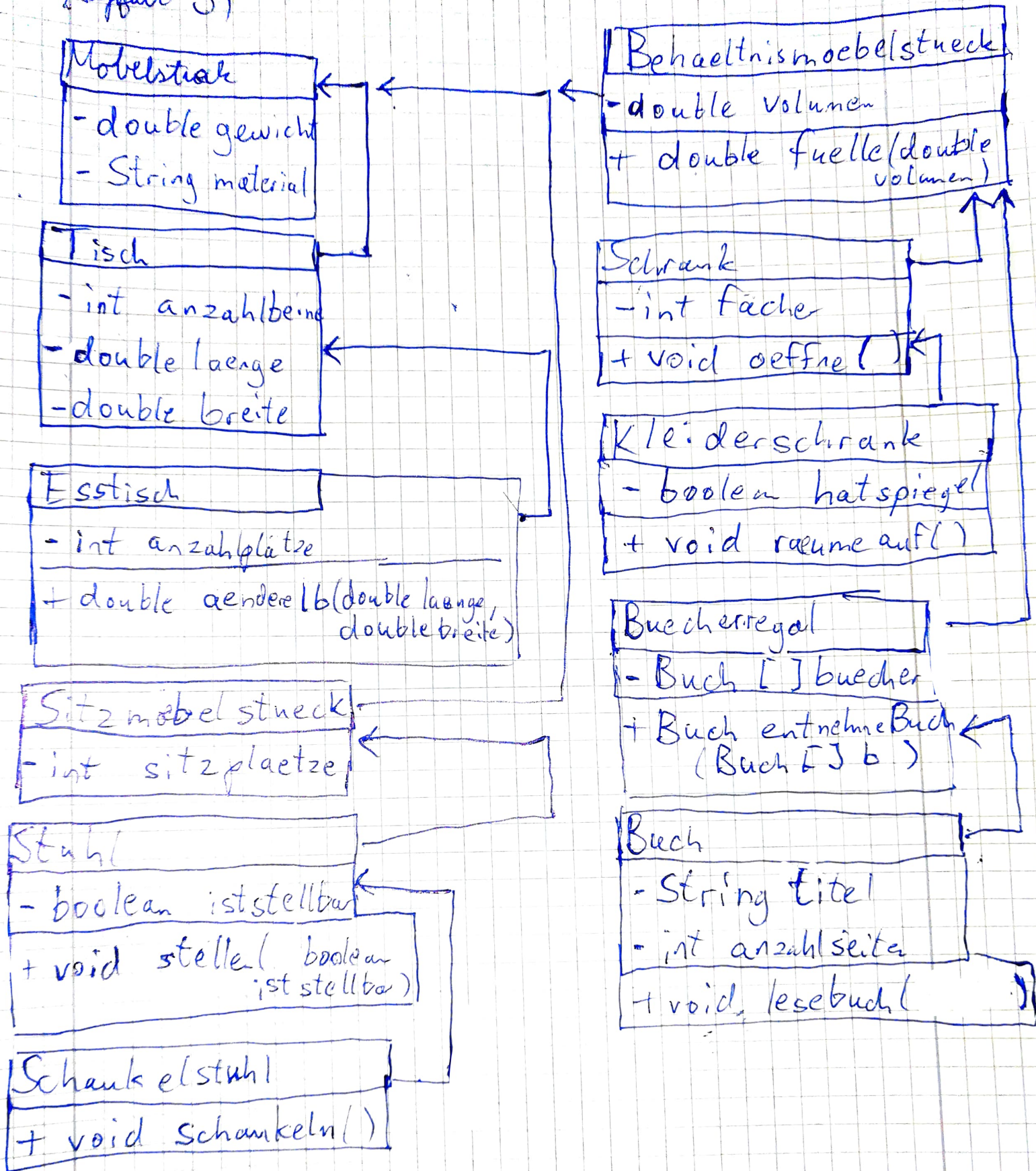
Übungsblatt 6 – 23.12.2020

Jannik Hiller – 422462

Davide Nico – 417862

Tutorium 14

Aufgabe 3)



Aufgabe 5

- a) (1) Konstruktoren:
1. A(I), da die ein int als Argument verwendet wird
 2. Object(), implizierter super() Konstruktor in A(I)
- Attribute:
- x = 0, Ursprungswert da x in A(I) vom Argument überdeckt wird
y = 0, da Neuzuweisung in A(I)
- println:
1. 0, Ausgabe des statischen Attributs x
 2. 0, Ausgabe des Attributs y
- (2) Konstruktoren:
1. A(), da es kein Argument gibt
 2. Object(), implizierter super() Konstruktor in A()
 3. A(I), explizierter this(int) Aufruf in A()
- Attribute:
- x = 1, in A(I) wird x vom Argument überdeckt und in A() um eins inkrementiert
y = 5, da Neuzuweisung in A(I)
- println:
1. 1, Ausgabe des statischen Attributs x
 2. 5, Ausgabe des Attributs y
- (3) Konstruktoren:
1. B(), da es kein Argument gibt
 2. A(), implizierter super() Konstruktor in B()
 3. Object(), implizierter super() Konstruktor in A()
 4. A(I), explizierter this(int) Aufruf in A()
- Attribute:
- A: x = 2, da in A() um eins inkrementiert
y = 4, da Neuzuweisung in A(I), $y -= x + 2 = 1 + 2$
B: x = 2.5, da in B() um eins inkrementiert
y = 1, unverändert
- println:
1. 2, Ausgabe des statischen Attributs x von A
 2. 2.5, Ausgabe des Attributs x von b
 3. 4, Ausgabe des verdeckten A-Attributs y von b
 4. 1, Ausgabe des Attributs x von b
- (4) Konstruktoren:
1. B(F), da der Konstruktor mit einem int-Argument aufgerufen wird, B(F) der einzige Konstruktor mit einem Argument ist und int impliziet zu float gecasted werden kann
 2. A(D), explizierter super(x) Konstruktor in B(F) (float wird zu double gecasted)
 3. Object(), implizierter super() Konstruktor in A(D)
- Attribute:
- A: x = 2, unverändert (statisch)
y = 11, da Neuzuweisung in A(D), $y += x = 3$ und Inkrementierung um 1 in B(F)
B: x = 1.5, unverändert
y = 1, unverändert

println: 1. 11, Ausgabe des verdeckten A-Attributs y von ab
 2. 1, Ausgabe des Attributs y von ab

- b)
- (1) A.f(IA), da direkt passende Signatur (int, A)
 - (2) A.f(DA), da long zu double gecasted werden kann
 - (3) B.f(IB), da direkt passende Signatur
(zur Compile-Zeit hat (B) ab den Typ B)
 - (4) A.f(DA), geerbte Methode mit passender Signatur
 - (5) B.f(IA), da A.f(IA) überschrieben wurde
 - (6) B.f(IA), zuerst wird A.f(IA) ausgewählt, da es die passendste Signatur hat, dann wird die Überschreibung ausgeführt.

Aufgabe 8

- g) Es ist nicht möglich die neue Gerade über die Attribute von Strahl zu verändern, da die verwendeten Punkte zwar identisch sind, aber nicht veränderbar sind, da sie 'final' sind (und die BigDecimal Werte der Punkte ebenfalls final sind).
- h) verlaengern() muss nicht überschrieben werden, da damit dann zu einer Geraden verlängert werden kann. (überladene Methode)
getStart() und getDir() müssen nicht überschrieben werden, da sie private sind und somit von Strecke nicht sichtbar.
- i)
1. Object.getClass aus Strecke.equals:34,
Aufruf einer öffentlichen Methode einer Überklasse
 2. Object.getClass aus Strecke.equals:34,
Aufruf einer öffentlichen Methode der eigenen Klasse
 3. Gerade.getPointA aus Strecke.equals:35,
Aufruf einer öffentlichen Methode einer Überklasse
 4. Point.equals aus Strecke.equals:35,
Aufruf einer öffentlichen Methode der eigenen Klasse
 5. Object.getClass aus Punkt.equals:33,
Aufruf einer öffentlichen Methode einer Überklasse
 6. Object.getClass aus Punkt.equals:33,
Aufruf einer öffentlichen Methode der eigenen Klasse
 7. BigDecimalUtility.equalValues aus Punkt.equals:32,
Aufruf einer öffentlichen statischen Methode einer anderen Klasse
 8. bigDecimal.scale aus BigDecimalUtility:8,
Aufruf einer öffentlichen Methode der eigenen Klasse