Tobias Grether, (205632), Tutorium 13
Lea-Noora Pdotschnig, (434855), Tutorium 13

Aufgabe 4 (Programmanalyse): (2+2+2+2+2+2+2+2+2+2 = 20 Punkte)

Lösen Sie die folgende Aufgabe ohne Einsatz eines Computers. Bedenken Sie, dass Sie in einer Prüfungssituation ebenfalls keinen Computer zur Verfügung haben.
Betrachten Sie das folgende kurze Programm:

```
public class B {

    private Integer i1;

    private int i2;

    private Double d;

    private float f;

    public B(int a, int b, int c, int d) {
        this.i1 = a;
        this.i2 = b;
        this.d = (double)d;
        this.f = c;
    }

    public B(Integer a, int b, Double c, float d) {
        this.i1 = a;
        this.i2 = b;
        this.d = c;
        this.f = d;
    }

    public Integer f(double x, int y) {
        return 11;
    }

    public int f(int x, float y) {
        return 12;
    }

    public int f(Double x, long y) {
        return 13;
    }

    public double g(Float x) {
        return 7.0;
    }

    public Float g(double x) {
        return 8f;
    }

    public static void main(String[] args) {
        B b1 = new B(1,2,3,4);
```

```
        System.out.println(b1.d);                          // a)
        System.out.println(b1.f(7d,8L));                   // b)
        System.out.println(b1.f(10d,17));                  // c)
        System.out.println(b1.f(5,6L));                    // d)
        B b2 = new B(b1.i1, 5, 6, 9);
        System.out.println(b2.f);                          // e)
        System.out.println(b2.f(b1.f,b1.i2));              // f)
        B b3 = new B(b2.i1, 14, 1.5, 16);
        System.out.println(b3.d);                          // g)
        System.out.println(b3.g(b1.i1));                   // h)
        System.out.println(b3.g(Float.valueOf(18)));       // i)
        System.out.println(b3.f(b2.g(19f), 21));           // j)
    }

}
```

Geben Sie die Ausgabe dieses Programms an, wenn die `main`-Methode ausgeführt wird. **Begründen Sie Ihre Antwort!** Ordnen Sie jeder Teilaufgabe die aufgetretenen Effekte zu und erklären Sie, warum gerade diese zu beobachten sind. Nehmen Sie dabei auch Bezug auf die Konstruktor-Aufrufe.

a) 4.0, da im Konstruktor d explizit gecastet wird.

b) 13, da durch Autoboxing die Methodensignatur (Double, long) erreicht wird.

c) 11, da die Methode f(double, int) die exakt richtige Methodensignatur hat.

d) 12, da dann der Long 6L zu einem float gecastet wird.

e) 6.0, da das Argument c (hier int 6) im Konstruktor explizit zu einem float gecastet wird

f) 11, da bei b1.f(float, Integer) der float zu einem Double gecastet wird und der Integer unboxt wird

g) 1.5, da im Konstruktor der Property d der Wert des Arguments c, hier 1.5, zugewiesen wird.

h) 7.0, da die Property i1 des Typs Integer automatisch zu einem Double gecastet wird.

i) 7.0, da der Integer 18 hier explizit zu einem Float umgewandelt wird. Somit erfüllt er exakt die Methodensignatur g(Double)

j) Durch Autoboxing wird 19f zu einem Float. Was dazu führt, dass der Methodenaufruf g(Float) den double 7.0 zurückgibt. Durch dieses double-Ergebnis wird dann die Methode f(double, int) aufgerufen, welche 11 zurück gibt.

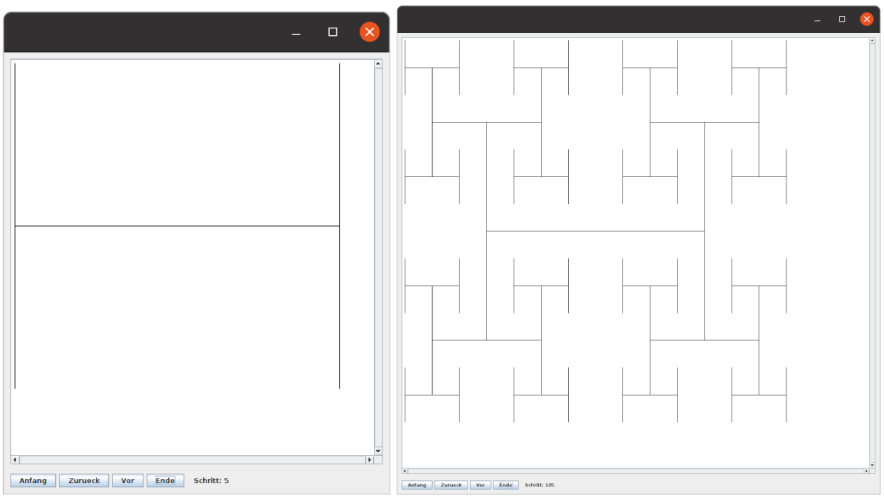

Abbildung 1: Beispiel: Programmausgabe für einen H-Baum der Tiefe 1 (links) und 3 (rechts)

Aufgabe 7 (Rekursion): (30 Punkte)

Auch in dieser Aufgabe soll eine fraktale Struktur mithilfe der Klasse `Canvas` gezeichnet werden. Hier geht es um sogenannte H-Bäume[1]. Dabei geht es um ein Fraktal, das wie folgt entsteht. Ein H-Baum der Tiefe 1 ist lediglich ein H. Ein H-Baum der Tiefe n ist ein H, bei dem an den vier Enden des Hs je ein H-Baum der Tiefe $n-1$ hängt, so dass die Mitte des Mittelstrichs des größten Hs des Teilbaums der Tiefe $n-1$ das Ende des aktuellen Hs berührt. Eine Ausgabe des Programms könnte für die Tiefen 1 und 3 wie in Abbildung 1 dargestellt aussehen.

Ihre Aufgabe ist, eine Methode `drawHTree(int size, int n)` zu implementieren, die solche Bäume zeichnet. Der Parameter `size` bestimmt die Größe des größten Hs des Baums, wobei die Größe die Länge der H-Striche angibt (anders als in den meisten Schriftarten soll hier der Mittelstrich des Hs genauso lang sein wie die Seitenstriche). Der Parameter `n` bestimmt die Tiefe des zu zeichnenden Baums. Ein einfaches Gerüst, um Ihre Implementierung zu testen, ist in `HTree.java` gegeben. Ändern Sie die Parameter des `drawHTree`-Aufrufs in der `main`-Methode nach Belieben um das Programm auf Richtigkeit zu prüfen. Die Größe der Bäume sollte in jedem Rekursionsschritt halbiert werden. Benutzen Sie für die Implementierung keine Schleifen, sondern nur Rekursion.

Die Klasse `Canvas` bietet neben den bereits aus Aufgabe 5 bekannten Methoden `rotate` und `drawForward` zusätzliche Methoden an. In dieser Aufgabe wird zusätzlich noch die Methode `moveForward` relevant, die genau wie `drawForward` die aktuelle Position verändert, jedoch nichts zeichnet. Außerdem wird das Methodenpaar `push` und `pop` bereitgestellt. Die Methode `push` speichert die momentane Konfiguration (Ausrichtung und Position des Zeigers) auf einem Stack, während `pop` die oberste auf dem Stack liegende Konfiguration wieder herstellt. Als Beispiel für die Funktionsweise sehen Sie unten eine beispielhafte Verwendung von `push` und `pop` für ein Objekt `c` vom Typ `Canvas`, wobei die Kommentare die jeweils aktuelle Zeigerposition angeben. Neu erstellte `Canvas`-Objekte `c` sind stets nach unten ausgerichtet.

Siehe S. 4

```
1  //Position: x:0,y:0 Ausrichtung: 0 Grad (nach unten)
2  c.push();
3  c.moveForward(10);
4  //Position: x:0,y:10 Ausrichtung: 0 Grad (nach unten)
5  c.push()
6  c.moveForward(10);
7  c.rotate(180);
8  //Position: x:0,y:20 Ausrichtung: 180 Grad (nach oben)
9  c.pop()
10 //Position: x:0,y:10 Ausrichtung: 0 Grad (nach unten)
11 c.pop()
12 //Position: x:0,y:0 Ausrichtung: 0 Grad (nach unten)
```

[1] https://de.wikipedia.org/wiki/H-Baum

Aufgabe 8 (Deck 5): (Codescape)

Lösen Sie die Missionen von Deck 5 des Codescape Spiels. Ihre Lösung für die Codescape Missionen wird nur dann für die Zulassung gezählt, wenn Sie Ihre Lösung vor der einheitlichen Codescape Deadline am Samstag, den 22.01.2022, um 23:59 Uhr abschicken.

```
public void drawHTree(int size, int n) {
    if (n == 0) return;

    c.push();
    c.rotate(-90);
    c.drawForward(size / 2);
    c.rotate(90);
    c.drawForward(size / 2);
    drawAndRotate(size, n);
    c.rotate(180);
    c.drawForward(size);
    drawAndRotate(size, n);
    c.pop();
    c.push();
    c.rotate(90);
    c.drawForward(size / 2);
    c.rotate(-90);
    c.drawForward(size / 2);

    drawAndRotate(size, n);
    c.rotate(180);
    c.drawForward(size);
    drawAndRotate(size, n);
    c.pop();
}

public void drawAndRotate(int size, int n) {
    c.push();
    drawHTree( size: size / 2,  n: n - 1);
    c.pop();
}
```