

1. Klausur 11/I (A)

Dauer: 2 Schulstunden

Hilfsmittel: --

Name: www.r-krell.de

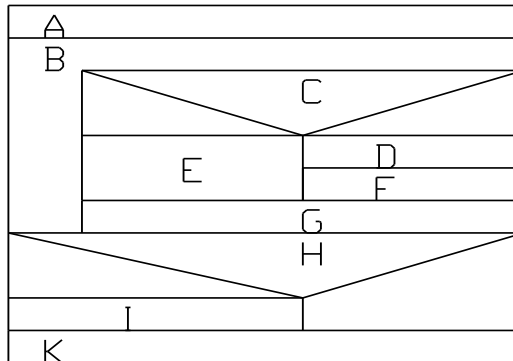
```

A; //1b)
if (B)
{
    C;
    if(D)
    {
        E;
    }
    else
    {
        F;
        G;
    }
    H;
}
else
{
    I;
}

```

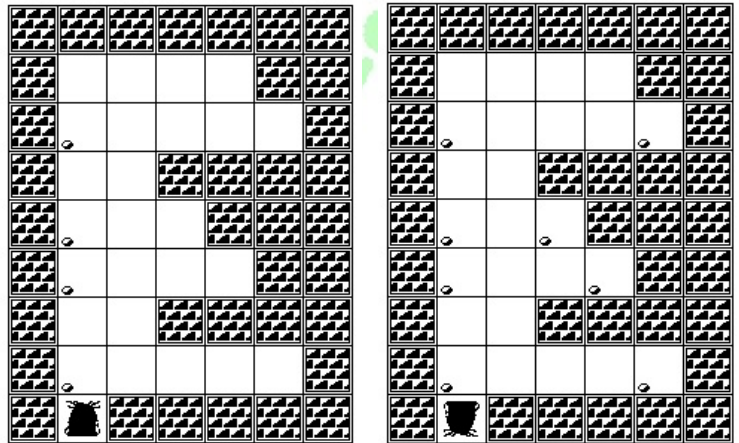
- * *Achte auf sorgfältige Darstellung mit nachvollziehbarem Lösungsweg!* *
- * *Kommentiere deine Programme!* *

1) Struktogramm und Java
Verwandle jeweils in die andere Form!



2) Hamster 1

Der Hamster startet unten links mit ganz vielen Körnern im Maul. Er geht zunächst hoch. Trifft er dabei auf ein Korn, so lässt er es liegen, geht aber in der selben Zeile soweit nach rechts wie möglich und legt dort noch ein Korn hin. Dann geht er wieder nach links und dort nach Möglichkeit weiter hoch. Zum Schluss kehrt er zu seiner Startposition am unteren Rand des Territoriums zurück. Schreibe das entsprechende Programm (mit mehreren Methoden, wie etwa *hHoch()*, *hNachRechtsUndZurück()*, *hRunter()*!)

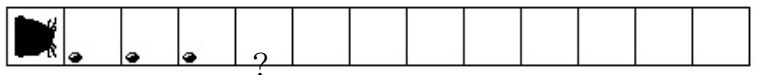


vorher

nachher

3) Hamster 2

Der Hamster startet auf einem freien Feld (oberes Bild). Er geht nach rechts und nimmt die einzeln hintereinander liegenden Körner auf. Gibt es keine Körner mehr (was der Hamster allerdings erst hinter den Körnern auf dem mit '?' markierten Feld bemerkt), so geht er von diesem freien Feld aus für jedes gefressene Korn noch zwei Schritte weiter – im Beispiel wegen 3 Körnern also insgesamt noch sechs Schritte und bleibt dann stehen (unteres Bild).



- a) Schreibe ein vollständiges Programm nur mit der *main*-Methode, in der alles passiert und in der eine Ganzzahl-Variablen definiert und benutzt wird.
- b) Schreibe zwei Methoden *hGeheUndZähle* (der Hamster geht nach rechts, sammelt dabei je ein Korn und liefert die Zahl der gesammelten Körner) und *hGeheDoppelt* (für die übergebene Ganzzahl *x* wird das Doppelte an Schritten – also $2 \cdot x$ Schritte – vor gegangen). Schreibe auch eine kurze *main*-Methode zum richtigen Aufruf der beiden Methoden mit Übergabe von *x*.
- c) Schreibe jetzt eine rekursive Methode *hRekursion()* (die sich selbst aufruft und keine Wiederholung enthält) um die Aufgabe zu erledigen. Bevor *hRekursion* aus der *main*-Methode aufgerufen wird, darf dort der Hamster zunächst einen Schritt vor machen (damit die Rekursion auf einem belegten Feld startet). Wenn nötig, darf auch nachher nochmal um einen Schritt korrigiert werden.

4 Fehlersuche

Gegeben ist nebenstehendes Programmstück. Leider wird es nicht richtig kompiliert, sondern enthält viele Fehler. Markiere die Fehler farbig hier auf dem Blatt und schreibe die berichtigte Fassung in dein Heft bzw. auf deinen Klausurbogen!

```
Hamster h = hamster.getStandardhamster;
```

```
Void Main;  
{  
  if Korn.Da()  
  }  
  h.nimm()  
  {  
  }
```

5 Programmanalyse

Wende das nebenstehende Programm auf das abgebildete Hamster-Territorium an. Zeichne den Nachher-Zustand des Territoriums in dein Heft, notiere, ob das Programm mit oder ohne Fehler endet und begründe kurz.!



vorher

