

Lösungen

AUFGABE 1: PROGRAMM SCHREIBEN (6 PUNKTE)

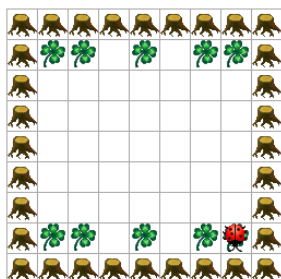
- 1 Punkt für While-Wiederholbedingung.
- 1 Punkte für Kleeblattaufnahme.
- 1 Punkt für korrektes Umlaufen der Pilze.
- 1 Punkte für korrektes Abbiegen an den Ecken des Rechtecks.
- 1 Punkt für korrekten else-Fall, das Geradeauslaufen.
- 1 Punkt für Verständlichkeit des Programms.

```
while (!kara.treeLeft()) {
    if (kara.onLeaf()) {
        kara.removeLeaf();
    }
    if (kara.mushroomFront()) {
        kara.turnLeft();
        kara.move();
        kara.turnRight();
        kara.move();
        kara.move();
        kara.turnRight();
        kara.move();
        kara.turnLeft();
    } else if (!kara.treeRight()) {
        kara.turnRight();
        kara.move();
    } else {
        kara.move();
    }
}
```

AUFGABE 2: PROGRAMM LESEN – PROGRAMM ERGÄNZEN UND LESEN (6 PUNKTE)

AUFGABE 2.1 (3 PUNKTE)

1 Punkt für leere Felder, 1 Punkt für Felder mit Kleeblättern, 1 Punkt für Position und Richtung von Kara.



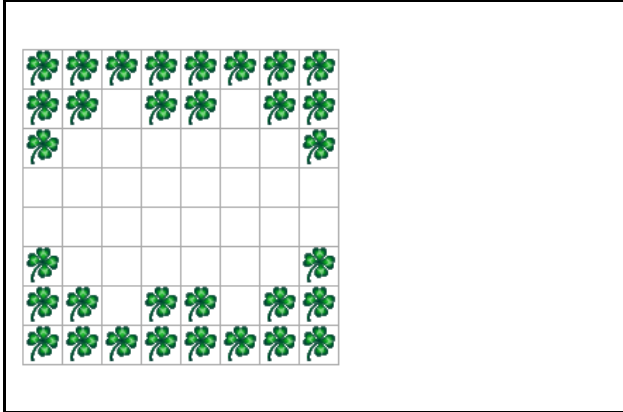
AUFGABE 2.2 (3 PUNKTE)

Die korrekte Zuordnung (4 oder 3 korrekt: 3 Punkt; 2 korrekt: 2 Punkte; ...): 1 – D, 2 – A, 3 – C, 4 – B

AUFGABE 3: JAVAKARA-PROGRAMM LESEN (6 PUNKTE)

AUFGABE 3.1 (3 PUNKTE)

1 Punkt, wenn ein Viertel korrekt; 2 Punkte, wenn Hälfte korrekt; 3 Punkte, wenn ganzes Bild korrekt.



AUFGABE 3.2 (2 PUNKTE)

Mögliche Lösung [2 Punkte - je einen Punkt]

legeBlaetter(...)	Die Methode legt in Spalte x Kleeblätter: anzahl Kleeblätter von oben her, anzahl Kleeblätter von unten her.
myMainProgram()	Das Hauptprogramm erfragt vom Benutzer eine Zahl. Es legt dann links und rechts von der horizontalen Mitte in Spalten Kleeblätter. Das Prozedere wird so oft wiederholt, bis jede Spalte der Welt befüllt ist.

AUFGABE 2.3 (1 PUNKT)

Mögliche Lösung [1 Punkt]

Die Welt, die das Programm zeichnet, ist horizontal symmetrisch: Von links und von rechts her wird jeweils das gleiche Muster in die Spalten gelegt.

AUFGABE 4: PROGRAMMIEREN ERKLÄREN (6 PUNKTE)

4 Punkte für Inhalt, 2 Punkte für Verständlichkeit, Beispiele.

Die wichtigsten Punkte, die erwähnt werden sollten:

- 1 Punkt: Was bedeutet zum Beispiel `void ganzeDrehung()` in Zeile 9? „Hier wird dem Computer ein neuer Befehl namens `ganzeDrehung` erklärt.“ – Methodendefinition. (Ohne Input, ohne Output.)
- 1 Punkt: Was bedeutet zum Beispiel `viertelDrehung()` in Zeile 11? „Hier sagen wir dem Computer, dass er den Befehl `viertelDrehung` ausführen soll.“ – Methodenaufruf. (Ohne Input, ohne Output.)
- 1 Punkt: Woher weiss der Computer, welcher Befehl als nächstes ausgeführt werden muss? „Der Computer arbeitet Zeile für Zeile. Muss er einen anderen Befehl wie zB `viertelDrehung` ausführen, schreibt er sich einfach auf einen „Spickzettel“, wo er anschliessend weiter machen soll.“ – Stack.
- 1 Punkt: Wo startet und endet die Programmausführung? „Es ist abgemacht, dass `myProgram` das Hauptprogramm ist, dass dort die Ausführung startet. Ist die Methode fertig abgearbeitet, endet auch die Programmausführung.“