

Methoden mit dem Hamster

Wir haben auf dem Skript-Seiten „Methoden in C#“ auf den ersten beiden Seiten erfahren, was Methoden überhaupt sind und wie man einfache Methoden in C# erstellt. Wir haben auch gesehen, dass Methoden einen Wert zurückliefern können, aber nicht müssen. Um etwas vertrauter mit Methoden zu werden, nutzen wir wieder den Hamster-Simulator.

Zur Erinnerung:

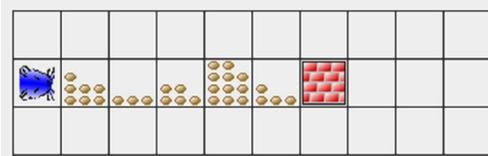
Der Hamster kann die folgenden Aktionen ausführen:

<code>vor();</code>	Der Hamster geht einen Schritt nach vorne.
<code>linksUm();</code>	Der Hamster dreht sich um 90° nach links.
<code>nimm();</code>	Der Hamster nimmt ein Korn auf.
<code>gib();</code>	Der Hamster legt ein Korn auf das Territorium.

Der Hamster besitzt folgende Sensoren, die alle `true` oder `false` zurückliefern:

<code>vornFrei();</code>	Prüft, ob der Hamster nicht vor einer Wand steht.
<code>kornDa();</code>	Prüft, ob auf dem Feld, auf dem der Hamster gerade steht, mindestens ein Korn enthalten ist.
<code>maulLeer();</code>	Prüft, ob der Hamster ein Korn im Maul hat.

Schon in den ersten Programmen mit dem Hamster haben wir uns z. B. daran gestört, dass sich der Hamster nur nach links drehen kann. Mit Hilfe von Methoden können wir jetzt eigene Hamster-Befehle, wie z. B. `rechtsUm()` erstellen. Ein anderer praktischer Befehl wäre `legeAlleKoernerAb()`, den wir uns im Folgenden ansehen:



Beim Hamster gibt es standardmäßig die `main`-Methode, die vom Simulator aufgerufen wird. Eigene Methode schreibt man (ähnlich wie in C#) vor oder nach der `main`-Methode, aber nicht in die `main`-Methode, also nicht zwischen den `{` und `}` nach `void main()`. Unsere neue Methode `legeAlleKoernerAb` hat wegen `void` keinen Rückgabewert.

Aufgabe 1:

- Welches Ergebnis liefert das rechts abgedruckte Programm?
- Kann man weitere Wiederholungen in eigenen Methoden zusammenfassen?

```

void legeAlleKoernerAb()
{
    while(!maulLeer())
    {
        gib();
    }
}

void main() {
    while(kornDa())
    {
        nimm();
    }

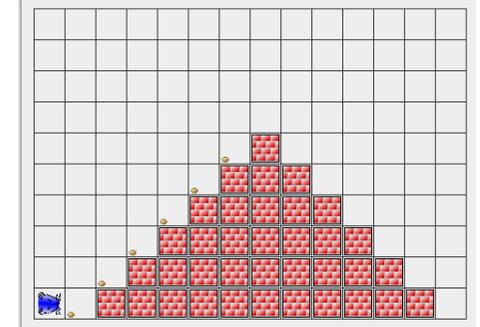
    while(vornFrei())
    {
        vor();
        while(kornDa())
        {
            nimm();
        }
    }

    legeAlleKoernerAb();
}
    
```

Aufgabe 2:

Auf dem Aufgabenblatt „while-Schleifen mit dem Hamster“ musste der Hamster in den Aufgaben 1 und 2 den rechts abgebildeten Berg erklimmen.

- Schreibe das Programm um, so dass jetzt Methoden sinnvoll verwendet werden.
- Ergänze das Programm um eine Methode `makeFreudensprung()`, die beim Erklimmen des Berggipfels aufgerufen wird. Beim Freudensprung bewegt sich der Hamster um ein Feld nach oben. Danach gibt der Hamstersimulator eine Erfolgsmeldung mit dem Befehl `schreib` aus und geht anschließend wieder um ein Feld nach unten.



Aufgabe 3:

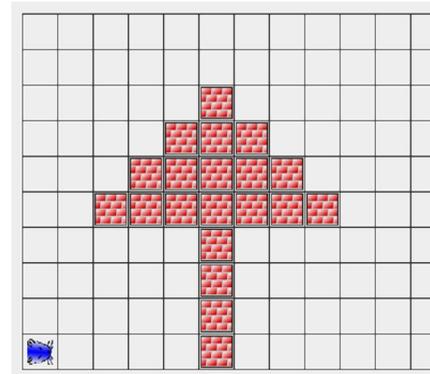
Schreibe das Programm zu Aufgabe 3 vom Blatt „while-Schleifen mit dem Hamster“ mit Hilfe von Methoden

Aufgabe 4:

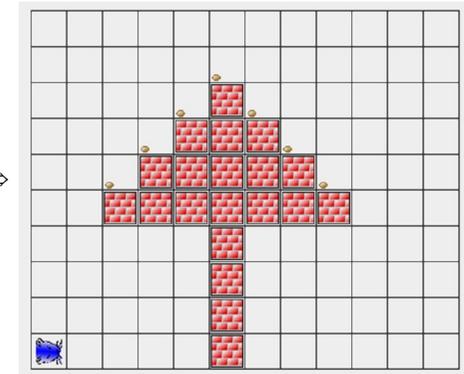
Schreibe das Programm zu Aufgabe 5 vom Blatt „while-Schleifen mit dem Hamster“ mit Hilfe von Methoden. Eine mögliche Lösung der alten Aufgabe findest du auf der Kursseite.

Aufgabe 5:

Ein Hamster soll z. B. einen Weihnachtsbaum oder Maibaum mit Schleifen (Körnern) verzieren. Das links abgebildete Territorium stellt die Ausgangssituation dar, das rechts die Endsituation nach Durchlauf des Programms:



⇒



Schreibe ein passendes Programm. Vergiss nicht, dem Hamster vor Programmstart ausreichend Körner ins Maul zu legen.