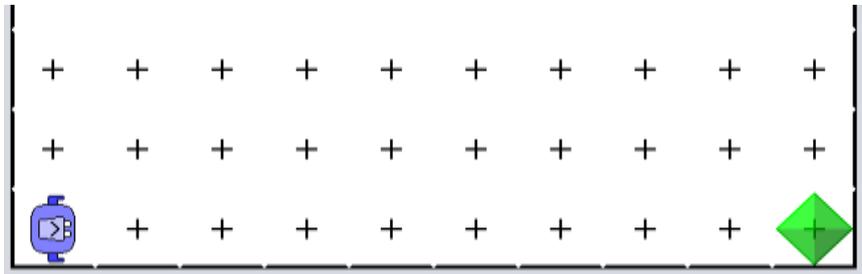


Folgende Information liefert uns die Aufgabe 1.1.2:

„Karel der Abbruchexperte entschärft eine Bombe am anderen Ende des Raumes und kehrt mit Stolz und Selbstvertrauen zurück. Er tat es ohne Codewiederholungen.“



```
void
defuseOneBomb ()
```

```
{
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    pickBeeper();
    turnAround();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    turnAround();
}
```

### 1. Eine Anweisung für Karel nachvollziehen

Schauen Sie sich den Code an. Beschreiben Sie in einem Satz, wie sich Karel bewegt und was er mit dem Beeper macht.

```
void defuseOneBomb ()
{
    repeat (9)
    {
        moveForward();
    }
    pickBeeper();
    turnAround();
    repeat (9)
    {
        moveForward();
    }
    turnAround();
}
```

„... er tat es ohne Codewiederholungen!“

Dieser Aufgabenteil wurde noch nicht erfüllt. Ganze 9 x hintereinander steht in unserem Code `moveForward()` und dies gleich an zwei Stellen.

Damit der Code insgesamt kürzer wird, benutzen wir das Schlüsselwort **repeat()**. Repeat steht für Wiederholung.

In der Klammer wird notiert, wie oft eine einzelne Anweisung oder ein Anweisungsblock (mehrere Anweisungen) wiederholt werden soll.

Die links dargestellte Anweisung erfüllt die gleiche Aufgabe wie die Anweisung von oben. Nur statt 21 Codezeilen hat man nun nur noch 7 (die Klammern muss man nicht in einzelne Zeilen packen; tut dies aber für die Übersichtlichkeit, deshalb zählen sie auch nicht mit).

### 2. Eine repeat-Anweisung für Karel nachvollziehen

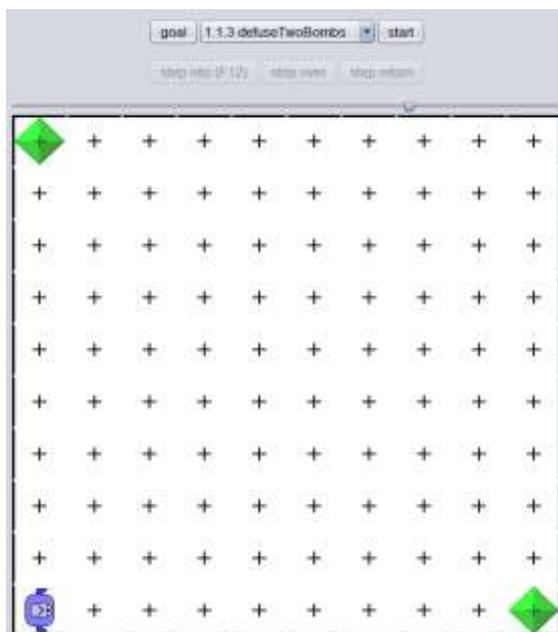
Der Code wurde nun nochmals etwas verändert.

Zeichnen Sie in das oberste Bild ein, wie der Zustand von Karels Welt nach Zeile 16 aussieht.

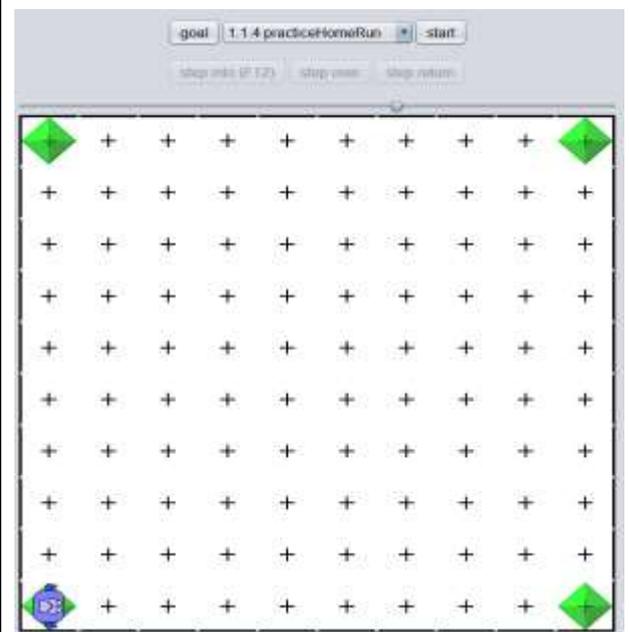
1. void defuseOneBomb()	9. pickBeeper();
2. {	10. turnAround();
3. repeat(8)	11. repeat(9)
4. {	12. {
5. moveForward();	13. moveForward();
6. dropBeeper();	14. }
7. }	15. turnAround();
8. moveForward();	16. }

**3. Karel will nun die Herausforderung steigern. Er weiß, wie man eine Bombe mit möglichst wenig Code entschärft. Schaffen Sie das auch für 2 Bomben? Entwickeln Sie einen Code unter Ausnutzung der Repeat-Methode. Startpunkt = Zielpunkt.**

**4. Karels Herz brennt für Baseball, aber er ist vom Zuschauen gelangweilt. Heute Abend wird er sich ins Stadion schleichen und seinen ersten "Home Run" durchführen. Pures Adrenalin! Startpunkt = Zielpunkt. Wer schafft es mit den wenigsten Codezeilen? Hinweis: In einer Repeat-Methode kann wiederum als Anweisung eine Repeat-Methode vorkommen.**



```
Void defuseTwoBombs()
{
```



```
Void practiceHomeRun()
{
```