**Die while-Schleife**

Die while-Schleife wiederholt eine oder mehrere Anweisungen solange (*engl. while*) eine Bedingung wahr ist. Die Bedingung steht in runden Klammern, sie muss als Ergebnis entweder true oder false liefern. Die Anweisungen, die wiederholt werden, stehen in geschweiften Klammern. Es werden immer alle Anweisungen ausgeführt bevor die Bedingung erneut geprüft wird. Ist die Bedingung nicht erfüllt, wird das Programm hinter der schließenden geschweiften Klammer fortgesetzt.

while ( *Bedingung* ) {

*Anweisung1*;

*Anweisung2*;

…

}

**Wichtig**: Eine while-Schleife steht immer *innerhalb* des Hauptprogramms, also zwischen den geschweiften Klammern nach main und am Ende der Datei.

1. Erste while-Schleifen (wird gemeinsam bearbeitet)
2. Schreibe ein Programm, mit dem der Hamster in seinem Gehege geradeaus bis zur nächsten Wand läuft und dann stehenbleibt. Teste es mit den Gehegen gehege01.geh bis gehege04.geh aus der letzten Stunde.
3. Welcher der folgenden Programmausschnitte lässt den Hamster bis zur nächsten Wand laufen und ein Korn von jeder Kachel einsammeln? Was passiert bei dem anderen Programmausschnitt? Teste beide Ausschnitte mit der Datei gehege03.geh.

while(vornFrei()){ while(vornFrei()){

vor(); vor();

nimm(); }

} nimm();



**Das weiß der Hamster über sich und sein Gehege**

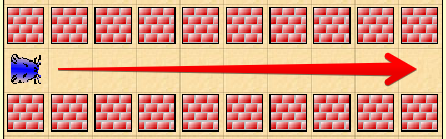
Der Hamster kennt drei Abfragen, um Informationen über sich bzw. sein Gehege zu bekommen:

vornFrei()

kornDa()

maulLeer()

Diese Abfragen liefern einen **bool'schen Wert** (boolean), also **wahr** (true/1) oder **falsch** (false/0). Anhand dieses Ergebnisses kann nun im Programm eine Unterscheidung getroffen werden, wie fortzufahren ist.

1. Bedingungen umkehren und kombinieren
2. Vorbereitung: Erstelle ein neues, leeres Gehege. Platziere den Hamster so, dass er möglichst weit geradeaus laufen kann. Gib dem Hamster Körner ins Maul, maximal so viele Körner wie Felder vor dem Hamster frei sind (im Bild: maximal 9 Körner).
3. Schreibe ein Programm: Solange der Hamster Körner im Maul hat soll er auf dem aktuellen Feld ein Korn platzieren und dann auf das nächste Feld vorrücken. Problem: Der Hamster soll etwas tun, wenn die Abfrage maulLeer() den Wert false liefert. Die Schleife wird aber ausgeführt, wenn die Bedingung true ist.

**Umkehren des Ergebnisses**

Möchte man das Ergebnis einer Abfrage invertieren (Logikfunktion NICHT), so muss man ein Ausrufezeichen (!) voranstellen.

1. Hat der Hamster mehr Körner im Mund als Felder vor ihm frei sind, so bricht das Programm mit einem Fehler ab wenn der Hamster gegen die Wand läuft. Probiere es aus!
2. Um das zu vermeiden sollte eine zweite Bedingung ergänzt werden: Der Hamster soll nur dann ein Korn ablegen und einen Schritt nach vorne machen, wenn

* er noch mindestens ein Korn im Maul hat (das Maul nicht leer ist)
* UND das Feld vor ihm frei ist.

**Bedingungen kombinieren mit UND**

Die Logikfunktion UND wird durch ein doppeltes Und-Zeichen (&&) dargestellt.

Gibt es mehrere Bedingungen, von denen nur eine erfüllt sein muss, so werden sie mit ODER kombiniert. Leider gibt es keine sinnvollen Beispiele dafür beim Hamstersimulator.

**Bedingungen kombinieren mit ODER**

Die Logikfunktion ODER wird durch zwei senkrechte Striche (||) dargestellt. Der senkrechte Strich befindet sich auf der Taste mit > und <. Um ihn zu schreiben muss entweder die Taste Alt Gr (rechts von der Leertaste) oder die Tastenkombination Strg + Alt zusätzlich gedrückt werden.

**Die if-Abfrage**

Die if-Abfrage führt Aktionen aus, wenn (*engl. if*) eine Bedingung wahr ist. Die Aktionen werden genau einmal ausgeführt, unabhängig davon ob die Bedingung anschließend immer noch erfüllt ist oder nicht mehr. Nach der Ausführung wird das Programm hinter der schließenden geschweiften Klammer fortgesetzt. Ist die Bedingung nicht erfüllt, wird das Programm sofort dort fortgesetzt.

if ( *Bedingung* ) {

*Anweisung1*;

*Anweisung2*;

…

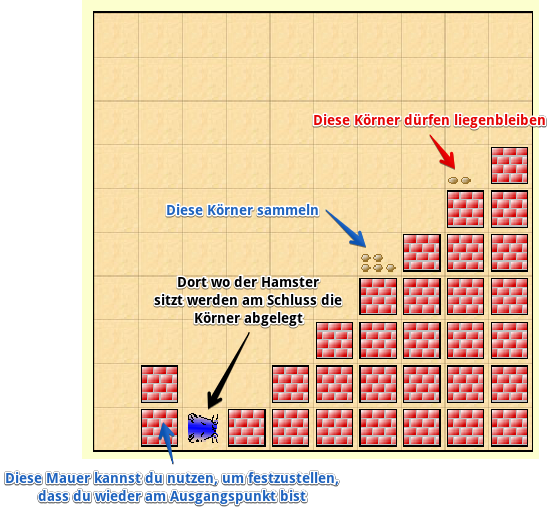
}

**Wichtig**: Eine if-Abfrage steht immer *innerhalb* des Hauptprogramms, oft auch innerhalb einer while-Schleife.

**Hinweis an die Könner:** else kommt noch – aber nicht mehr heute :)

1. if-Abfrage
2. Nutze das Gehege gehege04.geh vom letzten Mal. Der Hamster soll das Gehege entlanglaufen, aber von jedem Feld nur ein Korn nehmen – egal, wie viele Körner auf dem Feld liegen. Aber natürlich darf er nur versuchen, ein Korn zu nehmen, wenn dort überhaupt eines liegt, sonst bricht das Programm mit einem Fehler ab.
3. Jetzt kannst du auch die Gehege gehege05.geh und gehege06.geh vom letzten Mal mit Hilfe einer while-Schleife durchlaufen: ist das Feld vor dem Hamster nicht frei, muss er sich drehen bevor er einen Schritt nach vorne macht.
4. Übungen

Nutze while-Schleifen und if-Abfragen um die folgenden Aufgaben zu lösen. Manchmal gibt es mehr als einen Weg, der zum Erfolg führt, lasse dich also nicht davon verunsichern, wenn deine Klassenkameraden eine andere Lösung gefunden haben. **Speichere deine Lösungen zu diesen Aufgaben im Ordner H:\\_Einsammeln\_ !** (Gruppenlösungen sind innerhalb der Minigruppe erlaubt. Dann bitte angeben, dass es sich um eine Gruppenlösung handelt.)

1. Der Hamster soll ein gerades Gehege entlanglaufen bis er auf ein Feld trifft, auf dem Körner liegen. Dann soll er alle Körner auf diesem Feld einsammeln und dort stehenbleiben. Befinden sich keine Körner im Gehege, so soll er auf dem letzten Feld stehen bleiben.
2. Der Hamster soll ein gerades Gehege entlanglaufen. Dabei soll er alle Körner, die auf den Feldern liegen, einsammeln.
3. Der Hamster soll ein gerades Gehege entlanglaufen. Dabei soll er die Körner gleichmäßig im Gehege verteilen: ein Korn auf jedes Feld, am Ende umdrehen und zurücklaufen. Solange, bis er keine Körner mehr im Maul hat.
4. Der Hamster soll eine Spirale nach innen laufen. Dazu muss er immer geradeaus bis zur nächsten Wand laufen und sich dann nach links drehen. Das Ende der Spirale ist durch ein Korn markiert.
5. Der Hamster soll eine Spirale nach innen laufen. Allerdings fehlt nun das Korn, um das Ende der Spirale zu markieren.
6. Der Hamster soll eine Treppe nach oben steigen bis er Körner findet. Diese soll er einsammeln, zum Fuß der Treppe bringen und dort ablegen. Es müssen nur die Körner gesammelt werden, die auf dem ersten Feld liegen an dem der Hamster vorbeikommt. Allerdings sollen alle Körner, die auf diesem Feld liegen, gesammelt werden.

*Hinweis*: Für Teil a-c kannst du zum Beispiel die Datei gehege04.geh vom letzten Mal verwenden und die Körnerzahlen anpassen. Die Dateien für Teil d-f findest du in Ilias (als Zip-Datei, da Ilias sonst beim Herunterladen die Dateiendungen ändert).