

## **Bundeswettbewerb Informatik 2. Runde: Aufgabe 2**

### **1 Verwendete Begriffe innerhalb der Dokumentation**

*Turmrestaurant:* Das Turmrestaurant besitzt in der Mitte einen Zugang für Gäste. Dieser sollte mindestens einen Radius von einem Meter haben. Mit einem „Laufbereich“ von einem Meter für den Ober, kommt man auf einen minimalen Gesamtradius des Turmrestaurants von 2 Metern. Der Umfang dieser beiden Bereiche lässt sich berechnen mit:

$$U = 2 \cdot \pi \cdot r$$

Das ergibt einen Umfang von ca. 12,6 Metern. Man kann die minimale Anzahl der Sitzplätze im Turmrestaurant berechnen mit:

$$Anzahl\ Sitzplätze_{min} = \frac{U}{d_{Sitzplatz}}$$

Dabei steht d für den Platz, den ein Sitzplatz mindestens einnimmt. Ich habe diesen auf einem halben Meter festgelegt, so ergibt sich eine minimale Anzahl an Sitzplätzen von rund 25.

*Anzahl Sitzplätze (N):* Wie oben berechnet habe ich den minimalen Wert für N auf 25 festgelegt. Den Maximalwert von N habe ich auf 55 festgelegt, damit das Turmrestaurant bei der graphischen Ausgabe im Programm noch erkennbar ist. Letztendlich kann N jedoch beliebig groß sein.

*Sitzplatz:* Jeder Sitzplatz kann entweder belegt oder frei sein.

*Gruppe:* Eine Gruppe besteht aus mindestens 2 Gästen. Das Minimum habe ich anhand einer Umfrage festgelegt. In dieser waren 9 von 10 Personen der Meinung, dass eine Person keine Gruppe ist.

*Gruppe neben Gruppe:* Wird eine Gruppe neben einer anderen platziert, dann bedeutet dies, dass zwischen diesen kein freier Sitzplatz mehr ist.

*Freier Block:* Sind zusammenhängende freie Sitzplätze im Turmrestaurant.

*Belegter Block:* Sind zusammenhängende belegte Sitzplätze im Turmrestaurant.

*Tick:* Ist die Zeiteinheit in der Simulation.

*Verweildauer:* Eine Gruppe besitzt eine Verweildauer, die angibt, wie lange die Gruppe insgesamt im Turmrestaurant bleibt.

*(echte) Restzeit:* Die Restzeit ist in Ticks angegeben und gibt an, wie lange eine Gruppe noch im Turmrestaurant bleibt.

*Geschätzte Restzeit:* Ist die vom Ober angenommene Restzeit einer Gruppe. Um wie viel sie von der echten Restzeit abweicht, wird in Abhängigkeit der Erfahrung des Obers berechnet (siehe Abschnitt 2.2).

*Platzierung:* Der Ober befindet sich innerhalb des Turmrestaurants. Wird eine Gruppe links oder rechts neben einer anderen (platzierten) Gruppe platziert, dann geschieht dies aus der Sicht des Obers.

## 2 Lösungsidee

### 2.1 Simulation

Die Simulation läuft in Ticks ab. In einem Tick können mehrere Gruppen kommen.

Kommt eine Gruppe zum Turmrestaurant, so wird diese (wenn möglich) platziert, der Ober nimmt die Bestellung auf und schätzt danach die restliche Verweildauer der Gruppe ab. Ich habe festgelegt, dass dieser Vorgang keine Zeit benötigt. Bevor die Gruppe platziert wurde, ist dem Ober also nicht bekannt, wie lange die Gruppe im Turmrestaurant bleibt.

Wenn eine Gruppe geht, benötigt dies auch keine Zeit und die Sitzplätze können gleich wieder von anderen Gästen belegt werden.

### 2.2 Abschätzungsmethode des Obers

Der Ober kann nach der Platzierung der Gruppe anhand der Bestellung abschätzen, wie lange diese im Turmrestaurant bleibt. Bei jedem Tick schätzt der Ober die Restzeit der Gruppe erneut ab.

Hat beispielsweise die Gruppe gerade aufgegessen, so weiß der Ober, dass sie entweder bald die Rechnung verlangen oder aber noch ein Nachtisch bzw. etwas zu trinken bestellen. Die Gruppe wird aber nicht mehr länger als eine Stunde im Turmrestaurant sein.

Je höher die Erfahrung des Obers ist, desto genauer kann er abschätzen, wie lange die Gruppe noch im Turmrestaurant bleibt. Um dies zu simulieren wird für jede Gruppe eine maximale Abweichung zu deren echten Restzeit errechnet.

Eine Erfahrung von beispielsweise 60% bedeutet, dass die geschätzte Restzeit maximal um 40% von der echten Restzeit abweichen kann. Die maximale Abweichung wird berechnet mit:

$$Abweichung_{max} = Restzeit_{echt} \cdot (100\% - Erfahrung_{Ober})$$

Das Schätzen wird durch die Zufallsfunktion **Math.random()** simuliert welche einen zufälligen Wert zwischen 0 und 1 zurückgibt. Der Wert der geschätzten Restzeit lässt sich berechnen durch:

$$Restzeit_{geschätzt} = Restzeit_{echt} \pm Math.random() \cdot Abweichung_{max}$$

oder

$$Restzeit_{geschätzt} = (Restzeit_{echt} - Fehlerwert_{max}) + 2 \cdot Math.random() \cdot Abweichung_{max}$$

## 2.3 Platzierungsregeln

Für die Platzierung der Gruppen habe ich folgende Regeln aufgestellt:

Fall 1: es gibt nur einen freien Block:

- Fall 1.1: ist das Turmrestaurant leer, so kann die Gruppe irgendwo im Turmrestaurant platziert werden
- Fall 1.2: ist nur eine Gruppe im Turmrestaurant, so wird die Gruppe links oder rechts neben der anderen platziert (im Programm wird die Gruppe immer rechts daneben platziert)
- Fall 1.3: sind mehrere Gruppen im Turmrestaurant, so wird die Gruppe neben der Gruppe platziert, die länger im Turmrestaurant bleibt

Fall 2: es gibt mehrere freie Blöcke:

- Fall 2.1: passt die zu platzierende Gruppe genau in einen freien Block, so wird diese dort platziert; dabei wird die Gruppe in den zuerst gefundenen freien Block platziert
- Fall 2.2: trat keiner der obigen Fälle auf, wird die Gruppe in dem *kleinst möglichen* freien Block platziert, damit große freie Blöcke erhalten bleiben; die Gruppe wird dann wieder neben der Gruppe platziert, die länger bleibt

## 2.4 Der Ärger des Obers

Der Ober ärgert sich genau dann, wenn er für eine Gruppe genug freie Plätze im Turmrestaurant hat, diese aber nicht platziert werden kann, weil kein genügend großer freier Block zur Verfügung steht. Wenn die Anzahl der Gäste der Gruppe größer ist als die Gesamtanzahl der freien Plätze, so hat der Ober keinen Grund sich zu ärgern und schickt die Gruppe weg.

## 2.5 Der Streich:

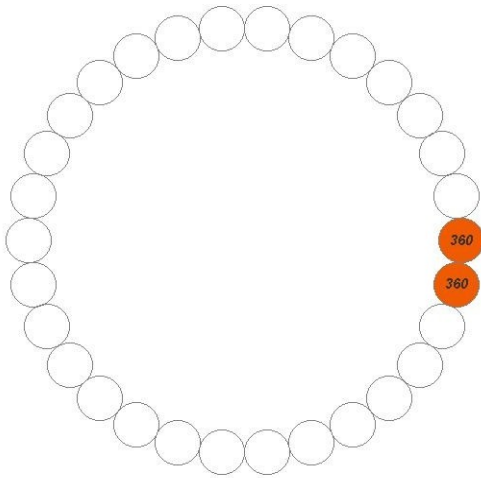


Abbildung 1

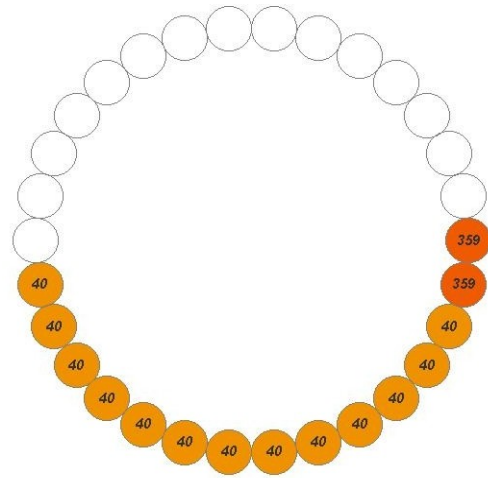


Abbildung 2

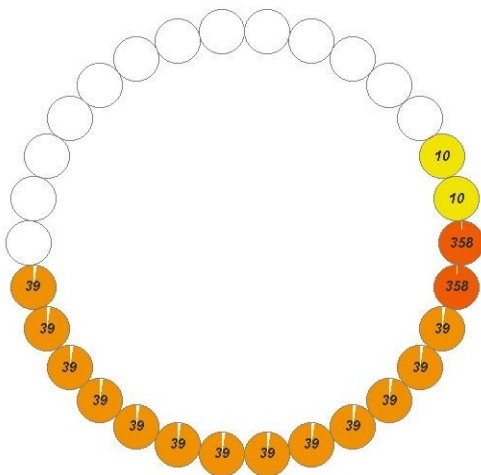


Abbildung 3

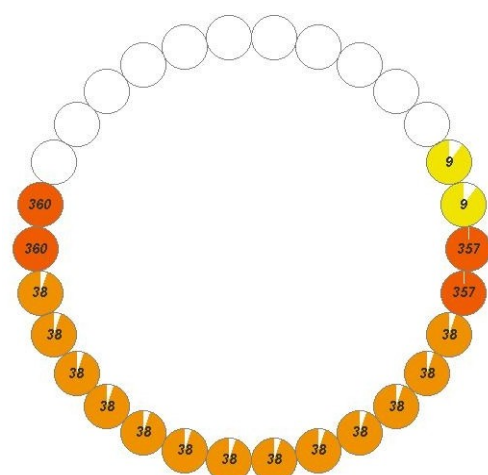


Abbildung 4

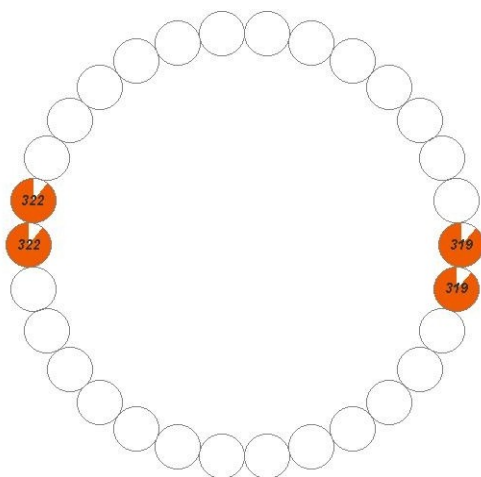


Abbildung 5



Abbildung 6

Bezeichnungen der Gruppen:

Bezeichnung	Anzahl der Schüler	Aufenthaltsauer
A	2	Sehr lang
B	13	Kurz
C	2	Sehr kurz
D	2	Sehr lang

Die Abbildungen zeigen wie der Streich bei einem Turmrestaurant mit 30 Sitzplätzen durchgeführt wird. Der Streich wird folgendermaßen durchgeführt:

Zunächst geht die Gruppe A in das Turmrestaurant. Da das Turmrestaurant leer ist, wird die Gruppe irgendwo platziert (Abb. 1).

Danach geht die Gruppe B in das Turmrestaurant. Die Anzahl der Schüler dieser Gruppe ist gegeben durch:

$$Anzahl_{GruppeB} = abrunden\left(\frac{N}{2}\right) - 2$$

Da nur eine Gruppe im Turmrestaurant ist, wird der Ober die Gruppe B neben der Gruppe A platzieren (Abb. 2).

Dann geht Gruppe C in das Turmrestaurant. Da der Ober abgeschätzt hat, dass die Gruppe A ziemlich lange bleibt, platziert er die Gruppe C neben der Gruppe A (Abb. 3).

Nun geht Gruppe D in das Turmrestaurant. Da die Gruppe B länger bleibt als Gruppe C wird die Gruppe D neben der Gruppe B platziert (Abb. 4).

Nachdem die Gruppe D platziert wurde sind keine Schüler mehr außerhalb des Turmrestaurants und es sind insgesamt 19 Schüler platziert worden. Wenn die Gruppen B und C gegangen sind, sitzen zwei 2er-Gruppen mit einem großen Abstand im Turmrestaurant (Abb. 5).

Nun gehen die Gruppen B und C als eine Gruppe in das Turmrestaurant und der Ober ärgert sich, was in dem **JFrame turmrestaurant** auch angezeigt wird (Abb. 6).

Die Anzahl der benötigten Schüler lässt sich berechnen mit:

$$\begin{aligned} Anzahl_{gesamt} &= Anzahl_{GruppeB} + 6 \\ Anzahl_{gesamt} &= abrunden\left(\frac{N}{2}\right) - 2 + 6 \\ Anzahl_{gesamt} &= abrunden\left(\frac{N}{2}\right) + 4 \end{aligned}$$

Der Streich funktioniert nur, wenn der Ober auch erfahren genug ist. Bei geringer Erfahrung des Obers werden die Restzeiten der Gruppen manchmal sehr ungenau abgeschätzt und der Ober platziert die Gruppen dann nicht wie oben beschrieben.

### 3 Programm-Dokumentation

Die Lösungsidee wurde von mir in der Programmiersprache Java umgesetzt. Mein Programm besteht aus folgenden Klassen: **TurmrestaurantMain**, **Turmrestaurant**, **Gruppe**, **Streich**.

Mein Programm besteht im Grunde aus zwei Teilen: Zum einen werden (je nach Einstellungen) zufällige Gruppen erstellt und der Methode **platziereGruppe** übergeben, die diese Gruppen dann platziert.

Die Platzierung der Gruppen nach den obigen Regeln sorgt dafür, dass sich der Ober möglichst selten ärgert. Dennoch gibt es Schwachstellen bei der Platzierung der Gruppen. Diese versuchen die Schüler zu nutzen und den Ober zu ärgern.

#### 3.1 Beschreibung der Klassen

Die Klasse **TurmrestaurantMain** enthält die **main**-Methode und erzeugt lediglich ein neues **Turmrestaurant**-Objekt. Die Klasse **Turmrestaurant** erzeugt eine **GUI** und enthält alle zur Simulation des Turmrestaurants wichtigen Variablen. In der Klasse **Turmrestaurant** sind Unterklassen, wie z. B. ein Zeichenpanel zur graphischen Ausgabe des Turmrestaurants enthalten oder aber **Listener** für **JButtons** und **JSlider**. Außerdem enthält die Klasse **Turmrestaurant** ein Objekt der Klasse **Streich** und drei **Array-Listen** der Klasse **Gruppe**. Die **Array-Liste gruppen** enthält die Gruppen, die zurzeit im Turmrestaurant sitzen. Die **Array-Liste gruppen\_ober** ist eine Kopie von **gruppen**, jedoch sind die Restzeiten dort von dem Ober abgeschätzt.

Die Methode **platziereGruppe** ist für die Platzierung der Gruppen verantwortlich.

Ein Objekt der Klasse **Gruppe** repräsentiert eine Gruppe im Turmrestaurant und enthält alle nötigen Informationen zum „Verwalten“ dieser, so z. B. die Anzahl der Gäste der Gruppe und die Restzeit dieser. Außerdem enthält ein Gruppen-Objekt einen Start- und Ende-Index, die anzeigen, wo die Gruppe im Turmrestaurant sitzt.

Die Klasse **Streich** „liefert“ die Gruppen für den Streich der Schüler und enthält alle wichtigen Informationen, um den Streich durchzuführen, so z. B. die Anzahl der Schüler, die für den Streich benötigt sind.

### 3.2 GUI

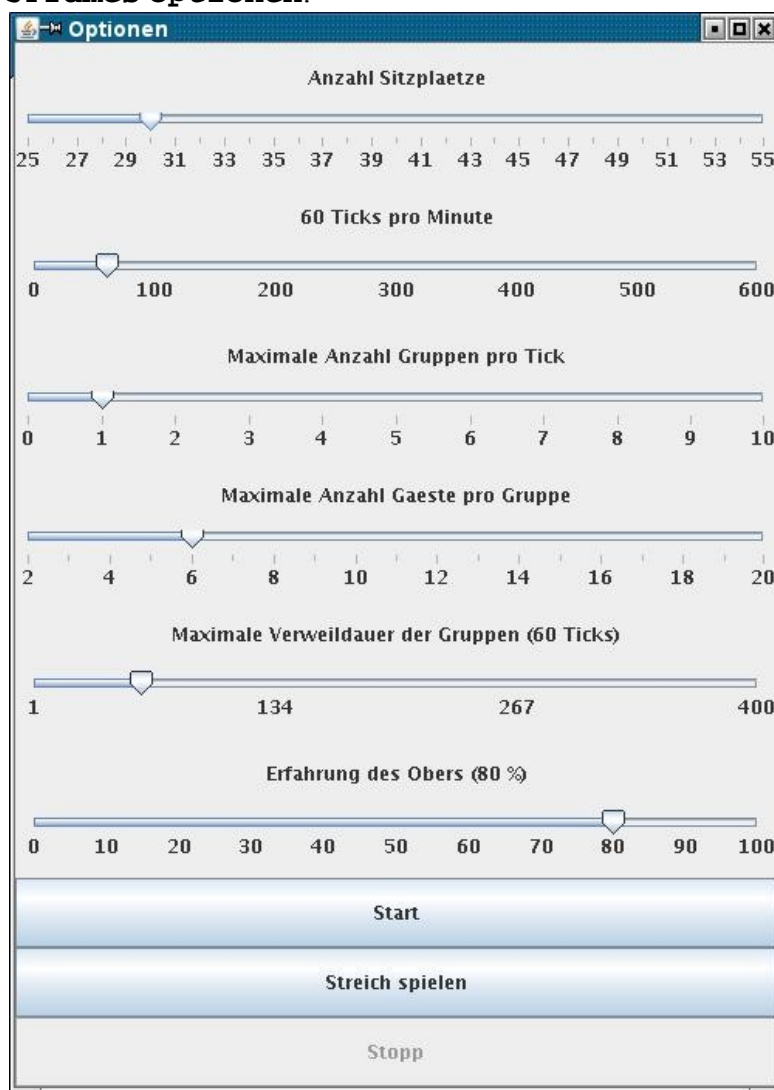
Die **GUI** besteht aus 2 **JFrames**.

Das **JFrame optionen** ist zum Einstellen der Eigenschaften der „ankommenden“ Gruppen, der Anzahl der Sitzplätze, der Simulationsgeschwindigkeit und der Erfahrung des Obers.

Durch Drücken des Start-Buttons werden zufällige Gruppen gemäß den Einstellungen erzeugt und platziert. Während der Simulation können die Einstellungen verändert werden. Beendet wird die Simulation durch Drücken des Stop-Buttons.

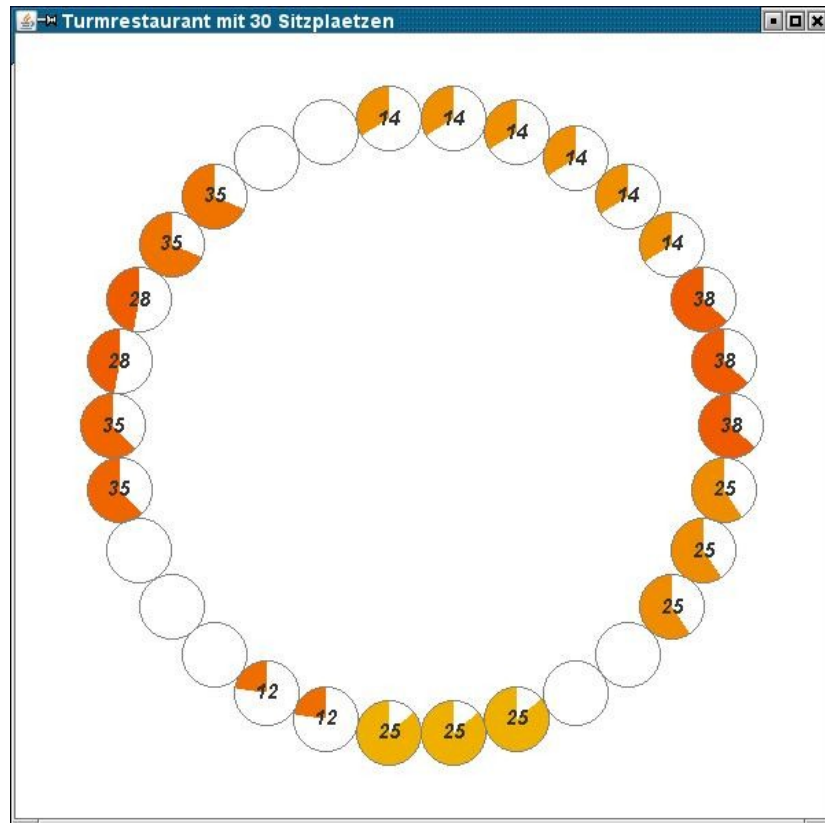
Möchte man, dass die Töchter des Obers den Streich spielen, so drückt man den „Streich spielen“-Button. Während der Streich gespielt wird kommen nur Schüler und keine „normalen“ Gäste.

Hier das Layout des **JFrames optionen**:



Das **JFrame** **turmrestaurant** zeigt das Turmrestaurant und die darin platzierten Gruppen. Die Sitzplätze des Turmrestaurants werden als Kreise dargestellt. Ist ein Sitzplatz belegt, wird der Kreis mit einer Farbe ausgemalt. Je dunkler die Farbe ist, umso größer ist die Verweildauer des Gastes bzw. der Gruppe. Die Restzeit des Gastes wird sowohl durch ein Label als auch durch die Größe des farbigen Kreissektors angezeigt.

Hier das Layout des **JFrames** **turmrestaurant**:

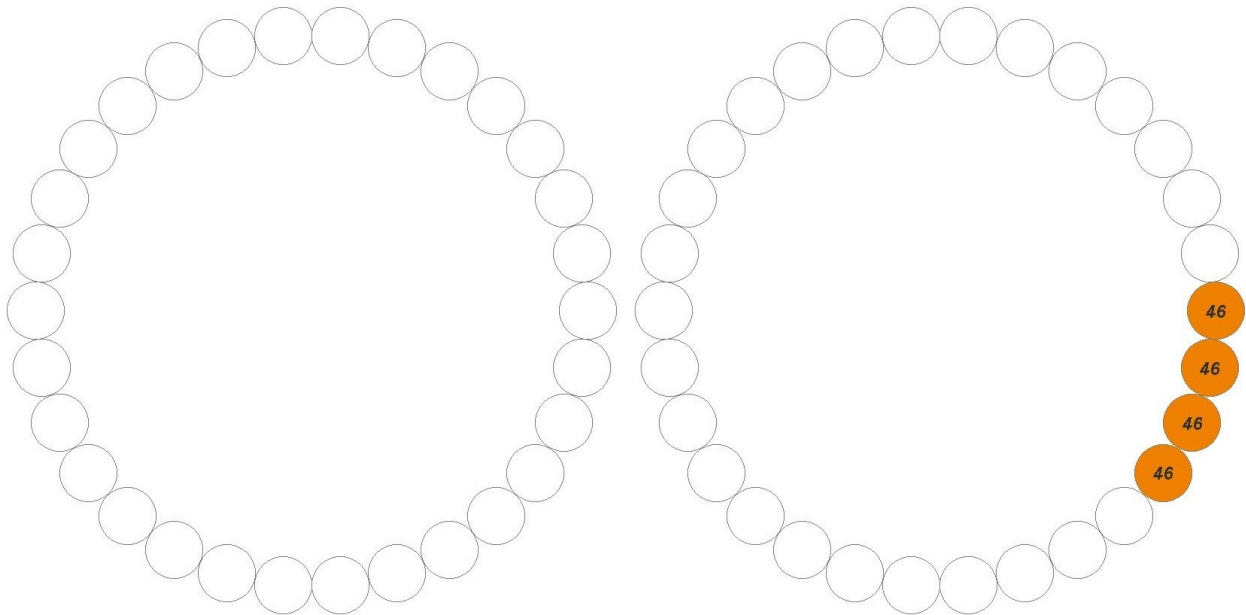




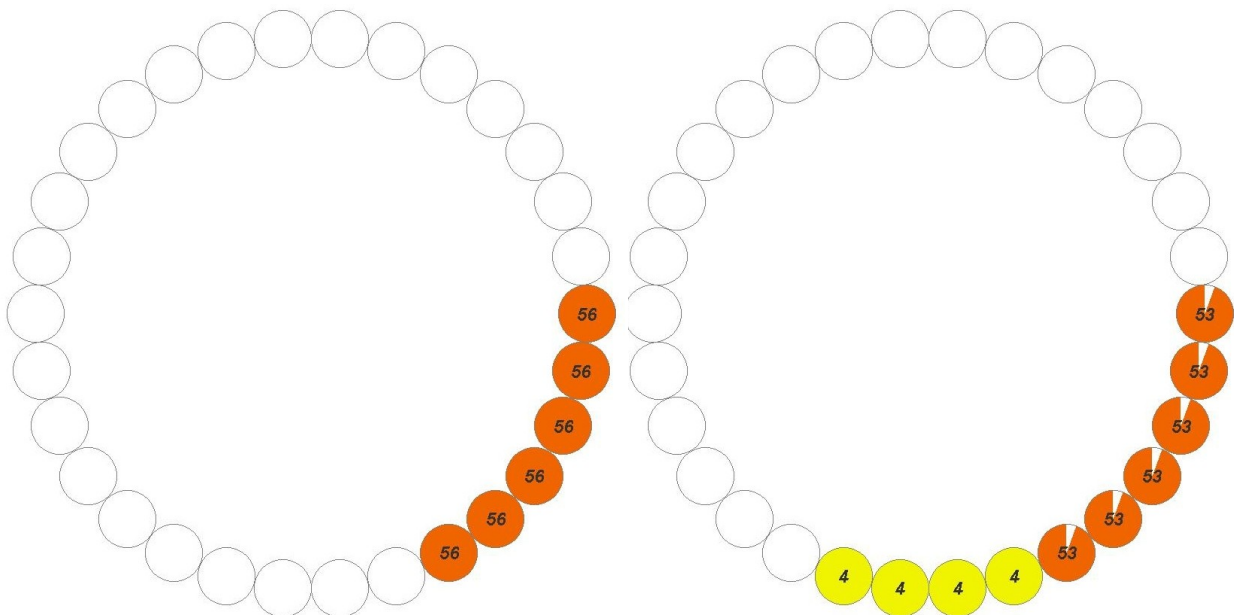
## 4 Programm-Ablaufprotokoll

### 4.1 Anwendung der Regeln:

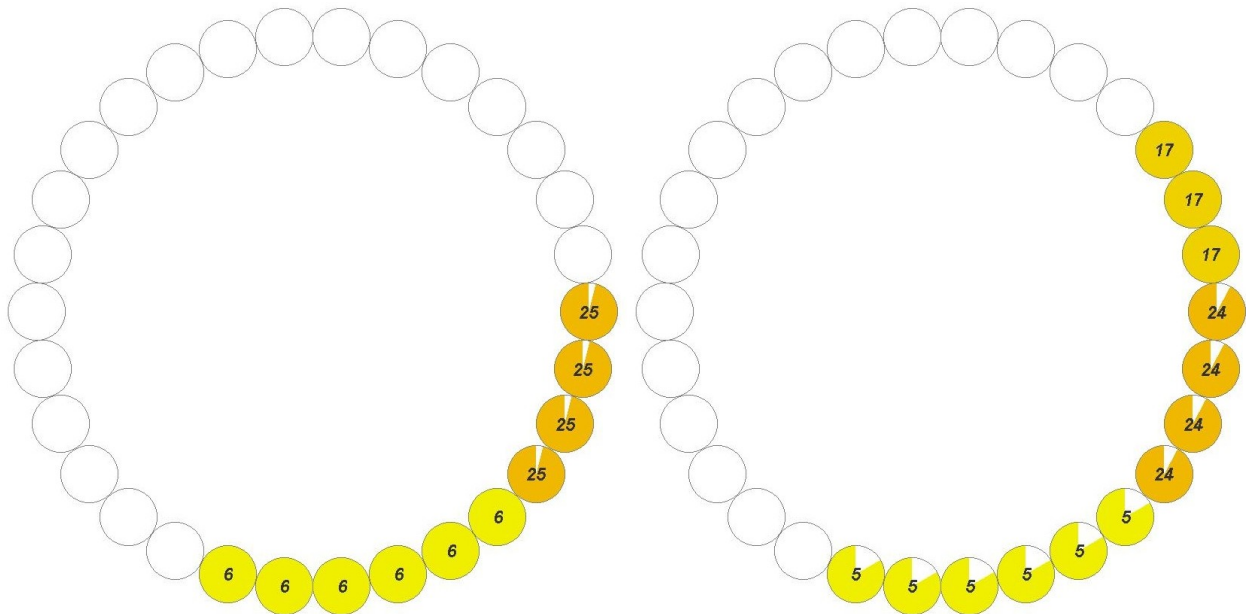
Ist das Turmrestaurant leer, wird die Gruppe irgendwo im Turmrestaurant platziert:



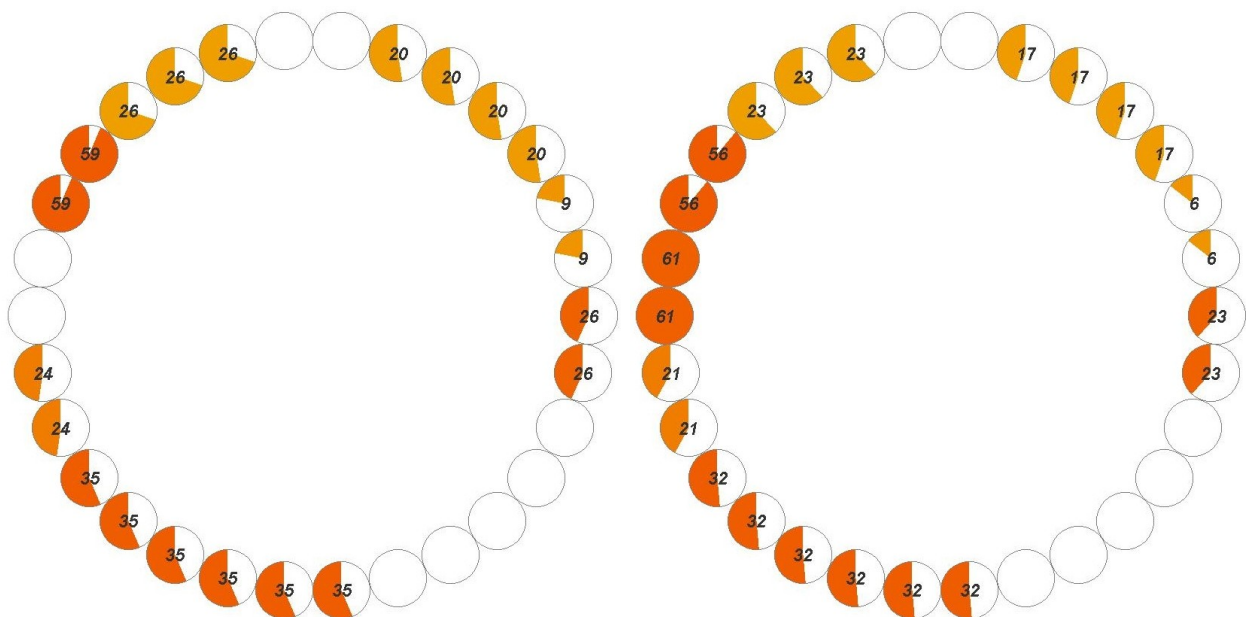
Ist nur eine Gruppe im Turmrestaurant, wird die Gruppe links oder rechts neben dieser platziert:



Gibt es einen freien Block und sind mehrere Gruppen im Turmrestaurant, wird die zu platzierende Gruppe neben die Gruppe, die länger im Turmrestaurant bleibt, gesetzt:



Gibt es mehrere freie Blöcke und passt die zu platzierende Gruppe genau in einen freien Block, wird die Gruppe dort platziert:



Ist keine der obigen Regeln angewendet worden, so wird die Gruppe in den kleinst möglichen freien Block platziert. Dadurch wird gewährleistet, dass große freie Blöcke erhalten bleiben:

