

Übung

Lottozahlen ziehen

|  |  |  |
| --- | --- | --- |
| GUS Deutschland GmbH Bonner Straße 172-176 50968 Köln  Telefon 0221 / 3 76 59-0 Telefax 0221 / 3 76 59-163  [**http://www.gus-group.com**](http://www.gus-group.com) | Autor(en): | Micha Hastrich |
|  | Version: | 1.0 |
|  | Versionsdatum: | 05.10.2016 |
|  | Copyright © GUS Deutschland GmbH | |

**Inhaltsverzeichnis**

1 Versionshistorie 3

2 Lottozahlen ziehen 4

2.1 Aufgabenstellung 4

2.2 Vorgehen 5

# Versionshistorie

| **Version** | **Datum** | **Bemerkung** |
| --- | --- | --- |
| 1.0 |  |  |

# Lottozahlen ziehen

Da wir jetzt die Tabellen Lotto\_<Benutzerkürzel> und Lottoziehungen haben, können wir eine wirkliche Tippabgabe und Ziehung simulieren.

Hierzu sollen mittwochs und samstags Zahlen gezogen werden und überprüft werden, ob und wenn ja wieviel Gewinn aus den Tipps entstanden ist. Dieser Gewinn wird dann in dem Tipp-Datensatz in der Lotto\_<Benutzerkürzel> gespeichert und das Flag gewonnen/nicht gewonnen wird gesetzt. Reihen ohne Gewinn werden mit negativem Gewinn belegt.

## Aufgabenstellung

Schreibt eine Schedulerklasse, welche Lottozahlen zieht und den Abgleich der Ziehung mit euren Tipps vollzieht. Diese soll dann auch die Gewinne direkt in die Tipps speichern und angeben, ob man etwas gewonnen hat.

Jeden Samstag und jeden Mittwoch erfolgt eine Ziehung.

Nehmt beispielhaft die Gewinnsummen einer bestimmten Ausschüttung und die reellen Kosten je Tipp zur Berechnung.

Zum Testen könnt ihr die Klasse so bauen, dass Sie immer die Gewinne zu der letzten Ziehung unabhängig von dem Datum auf alle eure Tipps berechnet. Die Klasse könnt ihr dementsprechend dann auch einfach einmalig ausführen und ihr müsst nicht auf Mittwoch/Samstag warten.

Damit ihr genug Werte zum Testen habt, baut ihr einen Funktionsaufruf auf dem Listpanel, welcher auch ohne das Entrypanel eine Tippreihe erstellt.

## Vorgehen

Funktionsaufruf zum Erstellen von Tippreihen

Erstellt einen EventHandler und programmiert hier die Methode exec.

In dieser werden Tippreihen generiert und in die Datenbank geschrieben.

Nutzt dazu den OSDCB und instanziiert ihn mit eurem Tabellennamen. Dadurch könnt ihr ihn später zum Einfügen in die Tabelle benutzen. Das Ganze findet ihr in diversen Beispielen:

|  |
| --- |
| // Ausgabe LIS-SATZ  OSDCB lisDCB = new OSDCB("LIS", this.con, this.sysInfo, false);  lisDCB.setNumeric("ISMAND", this.sysInfo.getCompany().toString());  lisDCB.setString("ISSB", this.sysInfo.getUser());  lisDCB.setString("ISID", "IS");  […]  sql = "INSERT INTO " + this.sysInfo.getSchema() + " LIS " + lisDCB.toSQLInsert();  stmt.executeUpdate(sql); |

Tipp: Eventuell müsst ihr euren Zähler von Hand auslesen (ID).

Das geht über die Klasse OSCounter.

Ziehen von Lottozahlen

Schreibt eine Schedulerklasse.

Schedulerklassen erben von der Klasse Job und benötigen die folgenden Methoden:

* Init
* Execute

Anbei findet ihr das Beispiel einer Schedulerklasse, die ihr mit Leben füllen könnt:

|  |
| --- |
| package de.guskoeln.products.os;  import java.sql.Connection;  import de.guskoeln.system.jDBV.system.SystemInfo;  import de.guskoeln.system.scheduler.Job;  /\*\*  \* Ziehung von Lottozahlen  \*/  public class Lottogenerator extends Job {  /\*\* Version-Tag für VCS \*/  public static final String SCM\_Version = "$Id$";  public void init(WfmProperties wfmProperties, Connection con, SystemInfo systemInfo) throws Exception {  super.init(wfmProperties, con, systemInfo);  }  /\*\*  \* Verarbeitung  \*/  public void execute(Connection con) throws Exception {  <Coding>  }  /\*\*  \* Abschluss der Verarbeitung  \*/  public void close() throws Exception {  super.close();  }  } |

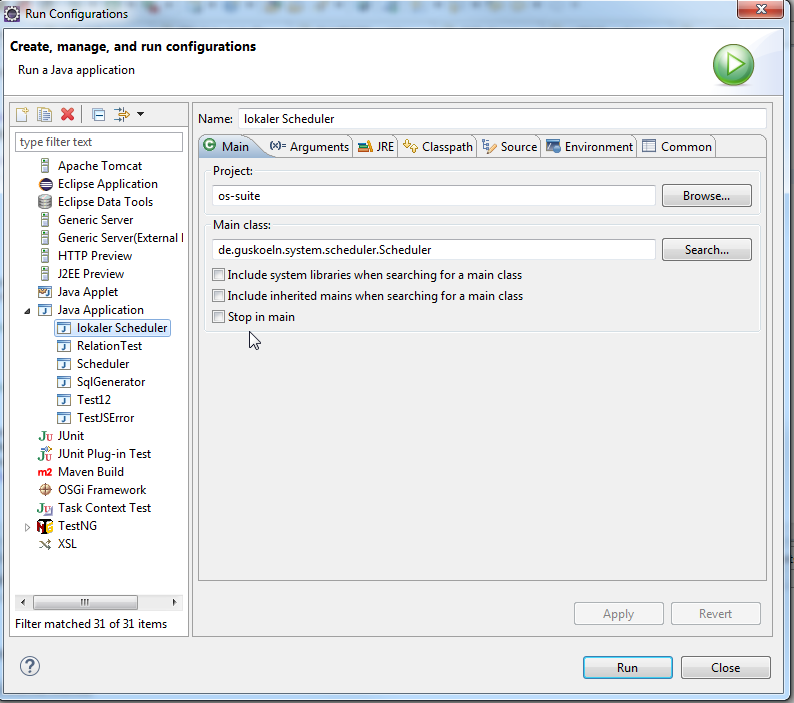
Wenn die Klasse fertig ist könnt ihr einen Schedulerjob definieren.

Die Konfiguration dafür findet ihr unter:

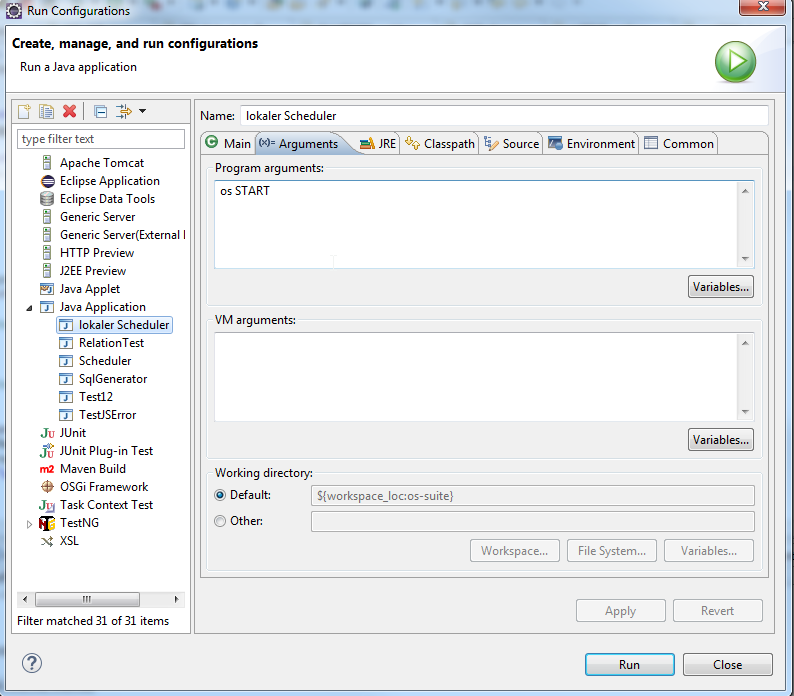
* Systemverwaltung
  + Allgemein
    - Scheduler

Erstellt hier einen neuen Job und stellt ein, dass er mittwochs und samstags ausgeführt wird.

Damit das Ganze bei euch lokal läuft, müsst ihr den Scheduler bei euch im Eclipse in den „Run Configurations“ erstellen und starten:



de.guskoeln.system.scheduler.Scheduler



os START