

# SPORT UPDATE – V1.0

```
-- change directory to program location such that relatively addressed resources can be found.
parse source      pgm
call directory filespec('L', pgm) -- change to the directory where the program resides

RexxApp~RexxApplication~new -- create Rexx object that will control the FXML set up
jrxApp=BSFCreateRexxProxy(RexxApp, "javafx.application.Application")
jrxApp~launch(jrxApp~getStage) -- launch the application, invokes "start"

::requires "BSF.CLS" -- get Java support

-- Rexx class defines "javafx.application.Application" abstract method "start"
::class RexxApplication -- implements "javafx.application.Application"

::method start
  use app -- fetch the primary stage (stage)
  setTitle("Welcome to Sport Update -- V.1.0")

  -- create an FXML loader (hence the prefix "bsf")
  fxmlUrl=.bsf~new("java.net.URL", "file:sport.fxml")
  -- use FXMLLoader to load the GUI graph from the file
  rootNode=bsf.loadClass("javafx.fxml.FXMLLoader")~load(fxmlUrl)

  scene=.bsf~new("javafx.scene.Scene", rootNode) -- create a scene for our document
  primaryStage~setScene(scene) -- set the stage to our scene
```

Business Programming 2  
Abschlussprojekt

18.06.2021

Thomas Kahr



# Uebersicht

- Fitnessdaten Tracker
- verschiedene Sportarten
- Sporteinheiten anlegen und Speichern
- vergangene Sporteinheiten ausgeben
- Zuruecksetzen
- Statistik plotten





# Komponenten

- GUI mittels JAVAFX
- Datenbank Sportarten  
SQLiteJDBC
- Charts mit JFreeCharts

# Aufbau des Programmes

## Hauptprogramm

Loginbereich als Dialog implementiert.

Hauptfenster als Scene mit primary Stage implementiert.

**sport.rxj**



**controller.rxj**



**database.rxj**



**ID, Sportart, Kilometer, Datum**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempus, lacus in dictum efficitur, dolor nisl dapibus turpis, nec dignissim lectus ipsum vitae velit.

**FXML/  
CSS**



## GUI Funktionalitaet


Speichern, Output,  
Zuruecksetzen, Statistik  
(Chart)

## GUI Bestandteile, Styling

Betrifft nur Hauptfenster

Hauptsaechlich ueber ID's  
angesprochen

Grosse Freiheit ueber CSS




# Funktionalitäten 1

- Speichern

im Controller wird auf den Click gewartet  
dann werden alle eingegebenen  
Informationen geprüft (ggf. Fehler) und  
alles in die Datenbank gespeichert.

- Zuruecksetzen

alle Labels, Textfelder und andere  
Elemente werden gecleared. Es wird  
ueberall das .nil-Objekt eingesetzt.  
zB. ob1~setText(.nil)



# Funktionalitäten 2

- Output

als erstes wird alles zurueckgesetzt.

Danach wird eine Datenbankabfrage durchgefuehrt und die letzten 10 Eintraege (neuester Eintrag 1) im GUI ausgegeben.

# Funktionalitaeten 3



- Statistik  
Sportart gewaehlt?  
Ja = Stat Sportart  
Nein = Stat alle Sportarten
- Chart wird erstellt und in eine PNG Datei gespeichert.  
(spaetere Verwendung fuer den Kuehlsschrank)
- Dialog wird gestartet und Grafik praesentiert.



```
::routine calcSeries public
use arg conn, sport

statement = conn~createStatement

series = .TimeSeries~new(sport)
input = .bsf~new("java.lang.String",sport)
query = statement~executeQuery("SELECT * FROM sporttable WHERE sport = '"sport"' ")
do while query~next
  km = query~getString("km")
  date = query~getString("date")
  year = date[1,4]
  month =date[6,2]
  day = date[9,2]
  series~add(.Day~new(day,month,year), km)
end
return series
```



# Probleme

- Wenig Dokumentation, viel Initiative erforderlich, trial and error
- Kurzer Projektzeitraum



An abstract graphic on the left side of the slide. It features a long, thin orange line that loops and swirls upwards. Below this, there are two smaller orange shapes: a semi-circular one with a white outline and a vertical, pill-shaped one, also with a white outline. The background is dark blue with a fine grid of small white dots.

# Fragen?