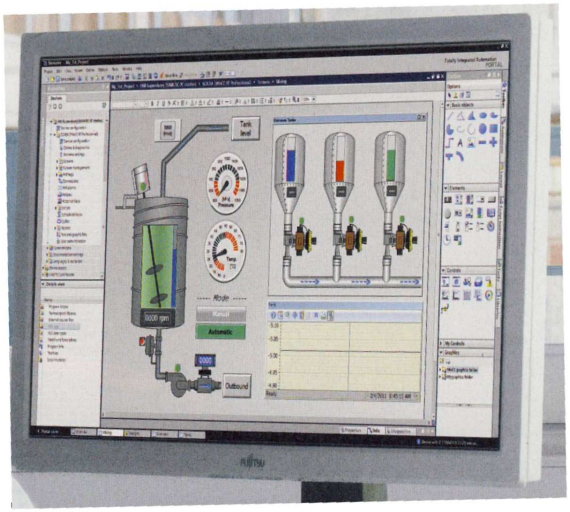
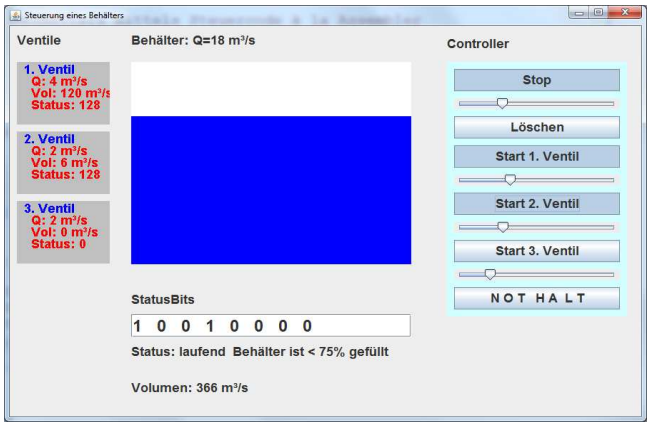
Behältersteuerung





Beschreibung

In der Mitte ist ein Panel-Behälter eingebaut, der durch Schalter gesteuert werden soll. Angeschlossen sind drei Ventile, die die Flüssigkeit aus dem Behälter lassen. Unter dem Behälter sind noch weitere grafische Elemente dargestellt:

- Beschriftungsfeld mit einer Editzeile
- Bemerkungs-Element
- Volumen-Element

Erläuterungen:

- Alle vier Elemente (Panel-Behälter, Beschriftungen etc.) sind View, Sichtweisen eines „realen“ Behälters in einer Fabrik. Das Programm soll die Steuerung simulieren.
- Auf der rechten Seite sind die Schalter und Slider eingefügt, mit denen man mit dem Programm kommunizieren kann.
- Auf der linken Seite sind drei Ventile dargestellt. Maximal könnten fünfzehn Ventile pro Behälter angeschlossen werden. Mit Q ist der Durchfluss, Menge pro Zeit gemeint. Die Abkürzung „Vol“ kennzeichnet die Summe des durchgeflossene Volumen.

Fast alle Befehle werden durch Bits und Bytes übermittelt:

Hinweis:
Auf einem Bus der das Ganze steuert, wurde hier verzichtet.

Ventil

Befehle an die Ventile:

- C_BEF_START = 0x10;
- C_BEF_STOP = 0x11;
- C_BEF_CLEAR = 0x1F;
- C_BEF_NOTHALT = 0xFF;

Statuswerte eines Ventils:

- C_RUN = 128;

Behälter

Befehle an den Behälter:

- C_BEF_START = 0x10;
- C_BEF_STOP = 0x11;
- C_BEF_CLEAR = 0x1F;
- C_BEF_NOTHALT = 0xFF;
- C_VENTIL_START = 0x20;
- C_VENTIL_STOP = 0x30;

Die Ventilbefehle sind durch ein High- und Low-Byte gekennzeichnet. Das High-Byte beschreibt die Funktion, das Low-Byte beschreibt die Nummer des Ventils, hier aber von eins beginnend.

Beispiel:

Kommando:	0x23	
Basiskommando:	0x20	also Starten eines Ventils
Lowbyte:	0x03	Nummer drei bzw. zwei in der Liste

Mit der logischen Verknüpfung "And" kann sowohl das High- als auch das Low-Byte bestimmt werden.

Statuswerte des Behälters:

- C_LEER = 1;
- C_LT_10 = 2;
- C_LT_25 = 4;
- C_LT_50 = 8;
- C_LT_75 = 16;
- C_LT_100 = 32;
- C_FULL = 64;
- C_RUN = 128;
- C_VENTILE_RUN = 256;

```
 // Check, welches Kommando angekommen ist
public void sendCommand(int command) {

   if (command == C_BEF_START) {
      setMaxVolumen(maxvolumen); // Alle View bekommen den aktuellen Wert
      ChangeStatus();
   }
   else if (command == C_BEF_STOP) {
      ChangeStatus();
   }
   else if (command == C_BEF_NOTHALT) {
      ChangeStatus();
      for (IGUIVentil ventil : listeVentile) {
       ventil.sendCommand(IGUIVentil.C_BEF_NOTHALT);
      }
   }
   if ( command == C_BEF_CLEAR) {
     volumen=0;
     ChangeStatus();
   }

      // Ventile, selektiere erst den Hauptbefehl, dann die Nummer
   if ( (command & 0xF0) == C_VENTIL_START ) {
     int nr = (command & 0x0F)-1;  // nun anschalten
     IGUIVentil ventil = listeVentile.get(nr);
     ventil.sendCommand(IGUIVentil.C_BEF_START);
   }
   else if ( (command & 0xF0) == C_VENTIL_STOP ) {
     int nr = (command & 0x0F)-1;  // nun ausschalten
     IGUIVentil ventil = listeVentile.get(nr);
     ventil.sendCommand(IGUIVentil.C_BEF_STOP);
   }
}
```

GUIBehaelterLabel

Methode setStatus:

Der Methode wird der aktuelle Status des Behälters als Parameter übergeben. Durch die Konstanten kann das Element nun die Status ermitteln und ausgeben. Hier wird die Ausgabe als Text vorgenommen.

```
public void setStatus(int status) {
  String sValue;
  if ( (status & Behaelter.C_RUN) == Behaelter.C_RUN)
     sValue="Status: laufend ";
  else
     sValue="Status: Stop ";

  if ( (status & Behaelter.C_LEER) == Behaelter.C_LEER)
    sValue+=" Behälter ist leer";
  else
    if ( (status & Behaelter.C_LT_10) == Behaelter.C_LT_10)
      sValue+=" Behälter ist < 10% gefüllt";
    else
      if ( (status & Behaelter.C_LT_25) == Behaelter.C_LT_25)
        sValue+=" Behälter ist < 25% gefüllt";
      else
        if ( (status & Behaelter.C_LT_50) == Behaelter.C_LT_50)
          sValue+=" Behälter ist < 50% gefüllt";
        else
          if ( (status & Behaelter.C_LT_75) == Behaelter.C_LT_75)
            sValue+=" Behälter ist < 75% gefüllt";
          else
            if ( (status & Behaelter.C_LT_100) == Behaelter.C_LT_100)
              sValue+=" Behälter ist < 100% gefüllt";
            else
              if ( (status & Behaelter.C_FULL) == Behaelter.C_FULL)
                sValue+=" Behälter ist voll";
              else {
                sValue+=" *********  Fehlerhafter Wert im Füllstatus-Bits  ****** ";
                CBasis.Message( "setStatus","Modul: GUIBehaelterLabel\nFehlerhafter Wert im Füllstatus-Bits");
              }
   setText(sValue);
}  // setStatus
```

GUIBehaelterTextField

Methode setStatus:

Der Methode wird der aktuelle Status des Behälters als Parameter übergeben. Durch die Konstanten kann das Element nun die Status ermitteln und ausgeben. Hier wird die Ausgabe als Bitfolge vorgenommen.

```
 // liefert  eins oder null, je nachdem, ob das Bits gesetzt ist
private int getBit(int value, int bit) {
   if ( (value & bit) == bit)
     return 1;
   else
     return 0;
 }


 // Nun die Bits auseinandernehmen
 public void setStatus(int status) {
    // Alternativ dazu wäre auch eine Schleife möglich
  int b7=0, b6=0, b5=0, b4=0, b3=0, b2=0, b1=0, b0=0;
  b0=getBit(status,1);
  b1=getBit(status,2);
  b2=getBit(status,4);
  b3=getBit(status,8);
  b4=getBit(status,16);
  b5=getBit(status,32);
  b6=getBit(status,64);
  b7=getBit(status,128);

  setText(  b7+"  "+b6+"  "+b5+"  "+b4+"  "+b3+"  "+b2+"  "+b1+"  "+b0 );
 }
```

Starten mit Eclipse

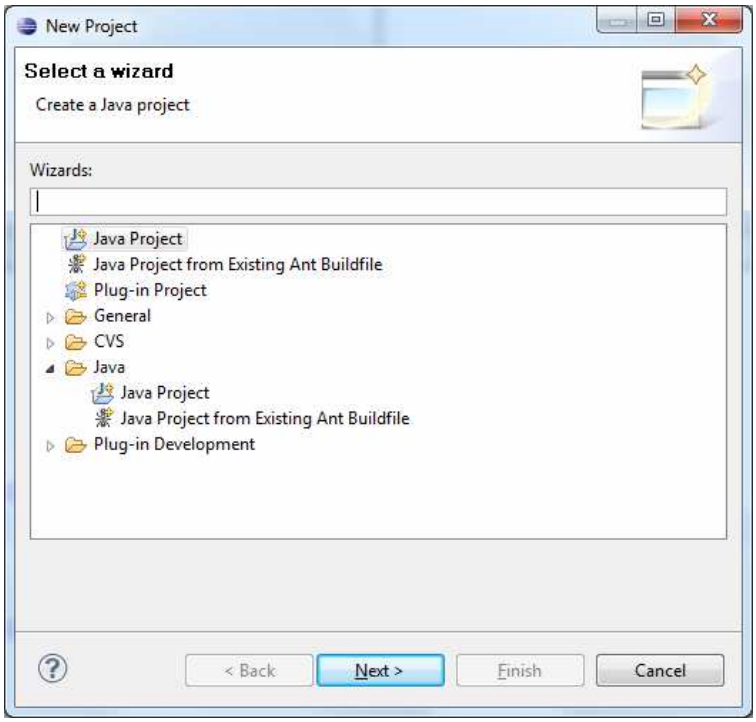
1) Download der Java-Datei, Anzeige im Browser

2) Starten von Eclipse

3) Neues Java-Projekt erstellen mit

   Menü "File" oder "Datei"

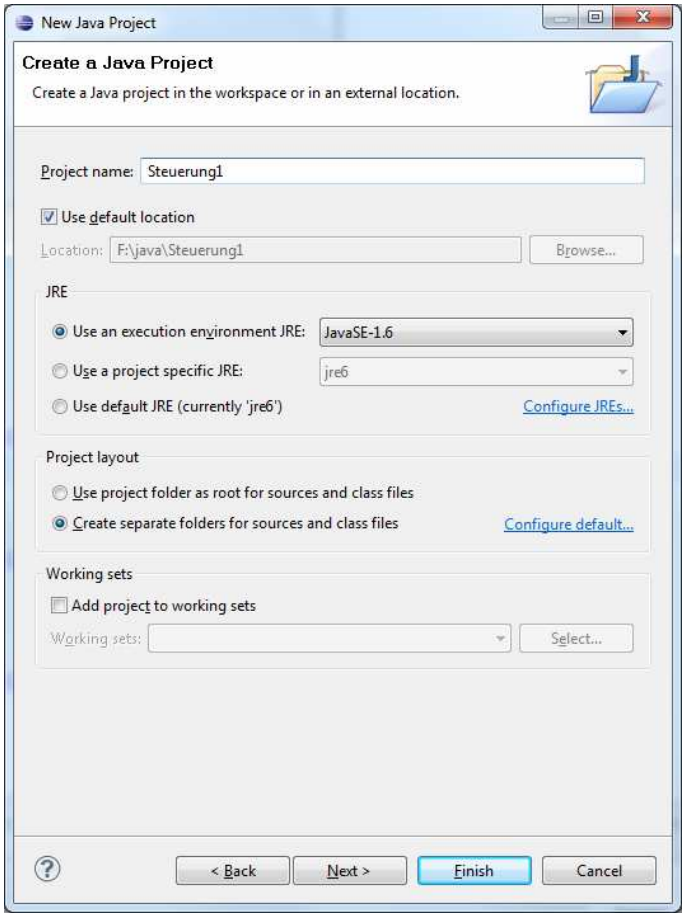
   Eintrag "Project" oder "Projekt"



   Auswahl im Dialog "Java Project" oder den Eintrag "other" wählen

   Schalter "Next" betätigen

   Es erscheint ein Dialog zur Projekt-Erstellung

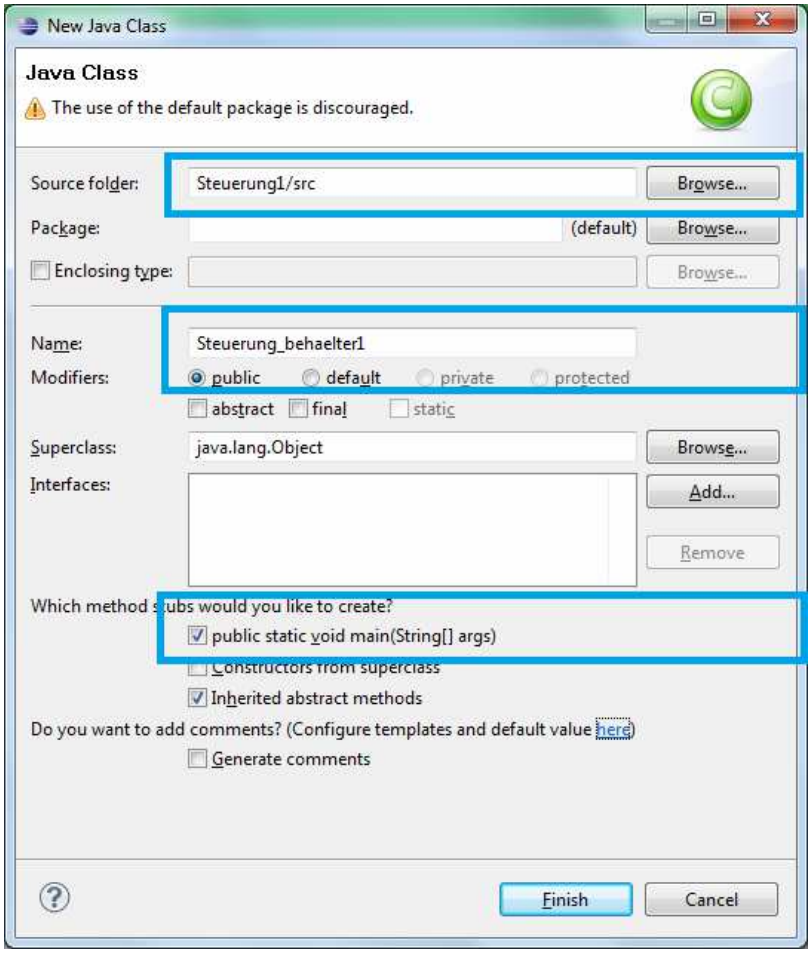


Eingabe des Projektname:
zum Beispiel: "Steuerung1"

Die Abfrage nach der Java-Perspektive bejahen

4) Das Projekt ist nun erstellt. Es fehlt noch die Java-Klasse.

Menü "File"
Eintrag "New"
Eintrag "Class"



Im Eintrag "Source-Folder" den korrekten Ordner auswählen

Im Eintrag "Name" den Text "Steuerung_behaelter1" einfügen.

Das Kontrollfeld, CheckBox, "public static void main" anklicken

Wechseln zum Browser mit der angeklickten Datei

Alles markieren mit Strg+A

Kopieren mit Strg+C

Wechseln zu Eclipse

Alles markieren mit Strg+A

Einfügen mit Strg+V

Hinweis:
Falls man einen anderen Namen für die Klasse eingetragen hat, muss man in der Java-Datei den Begriff " Steuerung_behaelter1" durch den Dateinamen ersetzen. Insgesamt muss man ihn dann viermal ersetzen.

5) Starten des Projektes mit Strg+F11