

## Aufgabe 5: SparKontoApplet 15.6. - 17.6

muss abgegeben werden mit dem Kommando `/proj/i4gdi/abgabe_A5`  
Schreiben Sie jede Teilaufgabe in ein eigenes File im Verzeichnis **Aufgabe5**

### Teilaufgabe 1: ( File SparKonto.java )

Erstellen Sie eine Klasse: **SparKonto**.

- Ein Objekt dieser Klasse soll die folgenden Bestandteile (Eigenschaften, Attribute) enthalten:

```
private String kontoNummer; // enthält die Kontonummer
private double kontostand; // enthält den aktuellen Kontostand
private int alter // Alter des Kontoinhabers in Jahren
```

- Der **Konstruktor** für ein neues SparKonto soll die folgende Signatur haben:

```
SparKonto( String kontoNr )
```

- Zusätzlich sollen Sie folgende Methode erstellen:

```
public double getKontostand() // liefert den aktuellen Kontostand zurück
public String getKontoNummer() // gibt die Kontonummer zurück
public int getAlter() // gibt das Alter des Kontoinhabers zurück
public void hatGeburtstag() // erhöht das Alter und schreibt dem Konto 3% Zinsen gut.
public boolean einzahlen( double betrag ) // falls betrag <= 0, nichts tun und false zurückgeben
// sonst betrag einzahlen und true zurückgeben
public boolean abheben( double betrag ) // falls betrag <= 0, nichts tun und false zurückgeben
// wenn ausreichend Geld auf dem Konto ist, betrag abheben und true zurückgeben
// reicht das Geld nicht aus (der Kontostand darf nicht negativ werden), nichts tun und false zurückgeben.
```

### Teilaufgabe 2: ( File SparKontoApplet.java )

Erstellen Sie ein Applet „**SparKontoApplet**“, mit dem Sie ein Sparkonto **mySparKonto** erzeugen mit Ihrer Matrikelnummer als Kontonummer. (Damit wir Testen können, bitte **kein private** im Applet verwenden)

Die Bank hat ihnen das Konto zur Geburt mit einem Guthaben von 10 EURO geschenkt. Zu jedem Geburtstag zahlt jemand 100 EURO auf das Konto ein . (Bisher wurde weder zusätzlich eingezahlt noch abgehoben)  
Führen Sie die Aktualisierung des Kontos mit den entsprechenden SparKonto-Methoden durch.

Die Oberfläche soll folgende Bestandteile in dieser Reihenfolge haben: (Grid-Layout mit 6 Zeilen und 2 Spalten)  
Vor jedem Feld soll ein Label stehen, der angibt, wozu das Feld dient.

dahinter jeweils

- Ein Feld, das die Kontonummer anzeigt
- Ein Feld, das den aktuellen Kontostand anzeigt.
- Ein Feld, in dem das Alter des Kontoinhabers steht
- Ein Feld, zur Eingabe eines Betrags, den Sie abheben oder einzahlen wollen
- Nebeneinander zwei beschriftete Buttons, einer zum Einzahlen und einer zum Abheben
- Ein Meldefeld, das Meldungen über Erfolg oder Misserfolg der letzten Aktion ausgibt

Berechnen Sie in der init-Methode das Guthaben auf Ihrem Sparbuch entsprechend Ihrem Alter unter Verwendung der Methoden aus der Klasse SparKonto. (einzahlen() und hatGeburtstag() ) und initialisieren Sie die Felder “Alter”, “Kontonummer” und “Guthaben” mit den aktuellen Werten Ihres Kontos.

Schreiben Sie im Applet zusätzlich folgende Methoden

```
protected void aktualisiereKontostand() // schreibt den aktuellen Kontostand in das Kontostand-Feld
protected void melde( String meldung ) // schreibt die “meldung” in das Meldefeld
protected void handleAbheben(). // soll den Betrag aus dem Eingabefeld vom Konto abheben und ins
Meldefeld “Betrag erfolgreich abgehoben” schreiben, falls das Guthaben auf dem Konto ausreicht (bitte nicht selbst rechnen) und der Kontoinhaber schon 18 ist. Sonst soll ausgegeben werden, “Konto ist gesperrt” oder “zu wenig Guthaben oder negativer Betrag”. Vergessen Sie nicht, den Kontostand zu aktualisieren.
Beim Anklicken des Abhebe-Buttons, soll im ActionListener genau diese Methode aufgerufen werden.
```

```
protected void handleEinzahlen() // Analog dem Abheben
```

Nennen Sie das html-File **SparKontoApplet.html**