



Aufgabenstellung: Sparplan erstellen in Java

Klasse:

Lege eine Klasse mit dem Namen **Anlage** an. Sie hat folgende Instanzvariablen:

- Anfangsbetrag
- Anlagejahr (zum Beispiel 2021)
- Zinssatz

Methoden:

1. Konstruktor

2. **Zinsen berechnen**

Hier wird berechnet wie viele Zinsen man für das laufende Jahr bekommt.

Zinsen = Kapital * Zinssatz

Die Zinsen werden zurückgegeben (= return Zinsen)

3. **Endkapital berechnen:**

Hier soll das Endkapital nach einer gewissen Anzahl von Jahren n berechnet werden, d.h. auf wieviel das Kapital durch die Zinsen angewachsen sind. Die Anzahl Jahre ist eine Importvariable.

Formel: Endkapital = Anfangskapital * (1 + Zinssatz)^{Anzahl-Jahre}

Java: $x^y = \text{Math.pow}(x,y)$;

Das Endkapital wird zurückgegeben (=return Endkapital)

4. **Sparplan erstellen-1:**

Der Anwender gibt den Zielbetrag vor und die Methode gibt für jedes Jahr eine Zeile mit `system.out.println` aus:

Jahr | Anfangsbetrag | Zinsen für das Jahr | Endbetrag.

Ist der Zielbetrag erreicht, bricht die Verarbeitung ab.

Hier soll eine **while (boolean expression) {statement(s)}** – *Anweisung verwendet werden.*

Zinsen pro Jahr = Anfangsbetrag * Zinssatz

5. **Sparplan erstellen (2)**

wie 1 nur mit `do {statement(s)} while (boolean expression)`;

6. **Sparplan erstellen (3)**

So wie Sparplan erstellen 1 oder 2, jedoch mit dem Unterschied, dass nicht das Zielkapital, sondern die Anzahl Jahre vorgegeben sind.

7. **Sparplan erstellen (4)**

Wie Sparplan 2, nur dass zusätzlich am Ende eines jeden Jahres noch ein festgelegter Betrag angelegt wird, dieser wird neben dem Zielbetrag durch den Anwender vorgegeben.



Schleifen

→ Anweisungen werden solange ausgeführt, bis eine Abbruchbedingung erfüllt ist.

in JAVA:

while (boolean expression) {statement(s)}

Die Bedingung (boolean expression) wird am Schleifenanfang geprüft. Die Anweisungen in der Schleife werden gegebenenfalls nicht ein einziges Mal ausgeführt. Die Anweisungen werden solange ausgeführt bis die Bedingung nicht mehr erfüllt ist.



do {statement(s)} while (boolean expression);

Die Bedingung (boolean expression) wird am Schleifenende geprüft. Die Anweisungen in der Schleife werden immer mindestens ein mal ausgeführt.

Methoden:

Diese könnt Ihr auch in euer Programm kopieren.

Damit die berechneten Zahlen nicht so viele Nachkommastellen haben (sieht hässlich aus, hier eine Methode zum Runden der Zahlen (aus dem Internet...))

```
public static double rundenEuro(double betrag) {
    //Runden - angepasst auf zwei Nachkommastellen
    double d = Math.pow(10, 2);
    return Math.round(betrag * d) / d;
}

public void sparplanErstellen(double zielbetrag)
{
    // Leerzeile und Überschrift drucken ....
    System.out.println();
    System.out.println("Jahr Anfangsbetrag Zinsen Endbetrag");
    // lokale Variablen deklarieren und initialisieren ...
    double zinsenNeu, betragNeu, endbetrag, anfangsbetrag;
    anfangsbetrag = betrag;
    int jahrNeu = jahr;
    // Zinsen und Endbetrag für das erste Jahr berechnen
    zinsenNeu= anfangsbetrag*zinssatz;
    endbetrag= anfangsbetrag+zinsenNeu;

    //Erstes Jahr unabhängig davon ob der Endbetrag erreicht ist ...
    System.out.println(jahrNeu + " "+ anfangsbetrag + " " + zinsenNeu + " " +
endbetrag);
    //Solange bis Zielbetrag erreicht
    while (endbetrag < zielbetrag)
    {jahrNeu =jahrNeu + 1;
    anfangsbetrag = endbetrag;
    zinsenNeu= rundenEuro(anfangsbetrag*zinssatz);
    endbetrag= rundenEuro(anfangsbetrag+zinsenNeu);
    System.out.println(jahrNeu + " "+ anfangsbetrag + " " + zinsenNeu + " " +
endbetrag); } }
```