Sichtbarkeit & statische Methoden

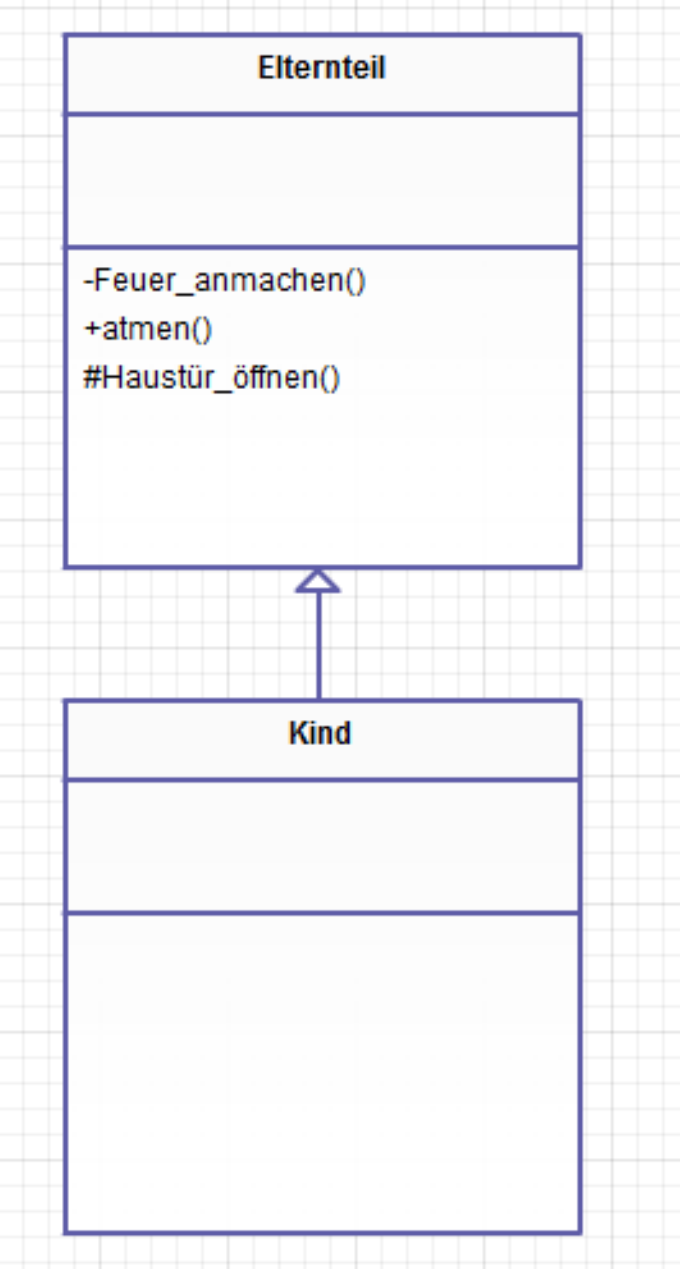
- Einsatz von Sichtbarkeit
- Einsatz statischer Methoden
- programmatische Realisierung
- 2 Beispielaufgaben

Nicht sichtbare Methoden

- Wollen Eltern bestimmte Methoden vor den Kindern „verstecken“ können diese als „private“ markiert werden
- Wollen Eltern den Kindern Zugriff auf Methoden erlauben, aber nicht der Außenwelt, müssen diese „protected“ sein
- Sollen Methoden für die Kinder und die Außenwelt aufrufbar sein, müssen diese public sein

Beispiel Sichtbarkeit

- Das Kind darf
  - Atmen und die Haustür_öffnen
- Kind darf nicht
  - Feuer_anmachen
- Keiner, außer Kind und Eltern dürfen Haustür_öffnen
- Nur Elternteil darf Feuer anmachen



Statische Methoden

- Normalerweise können Methoden nur aufgerufen werden, wenn eine Instanz der Klasse erzeugt wurde

```
public class Auto
{
	float aKilometerzaehler;
	public void fahren(float aKilometer)
	{
		Akilometerzaehler=
		aKilometerzaehler + aKilometer;
	}
}
```

```
public class Startprogramm
{
	Public static void main(String[] args)
	{
		Auto meinAuto=new Auto();
		meinAuto.fahren(10);
	}
}
```

Statische Methoden

- Statische Methoden können jederzeit aufgerufen werden, auch wenn noch kein Objekt erzeugt wurde

```
public class Auto
{
      ….

      public static float meilen2Km(aMeile)
      {
            return aMeile*1.6;
      }
}
```

```
public class Startprogramm
{
      Public static void main(String[] args)
      {
            float kilometer=
            Auto.meilen2Km(1);
      }
}
```

- Statische Methoden werden oft als kleine Hilfsmethoden genutzt, die von einzelnen Objektinstanzen unabhängig sind

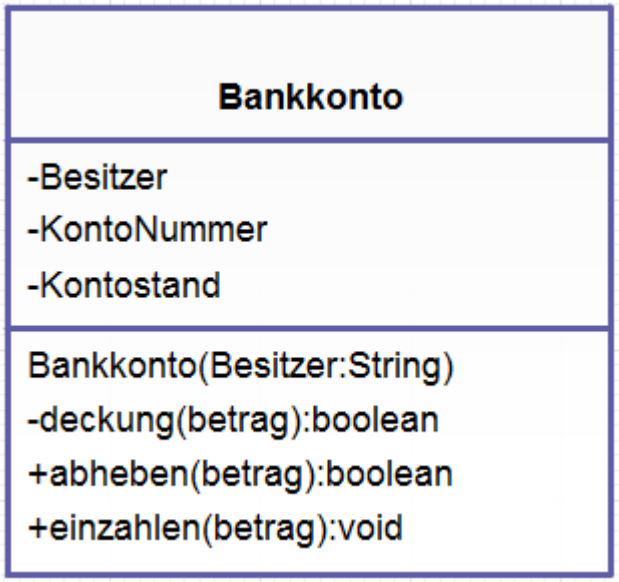
Task Bankkonto

- Bankkonten sollen verwaltet werden
- die Klasse Bankkonto verfügt über Zustandsvariablen (die nur durch die eigenen Verhaltensmehoden geändert werden dürfen)
  - den Besitzer
  - eine Kontonummer
  - den Kontostand

Task Bankkonto

Verhaltensmethoden

- den Konstruktur mit Parameter Besitzer
  - eine laufende Nummer soll automatisch vergeben werden für jedes neu eröffnete Bankkonto



- eine private Methode deckung(betrag:float):boolean
  - die Methode gibt false zurück, wenn nicht genug betrag auf dem Konto verhanden ist
- abheben(betrag:float):boolean
  - es darf nicht mehr abgehoben werden, als Geld zur Verfügung steht (hier wird die Methode deckung benutzt), bei erfolgreichem abheben wird true ausgeben, sonst false
- einzahlen(betrag:float):void

Task Bankkonto

Die Klasse Bank

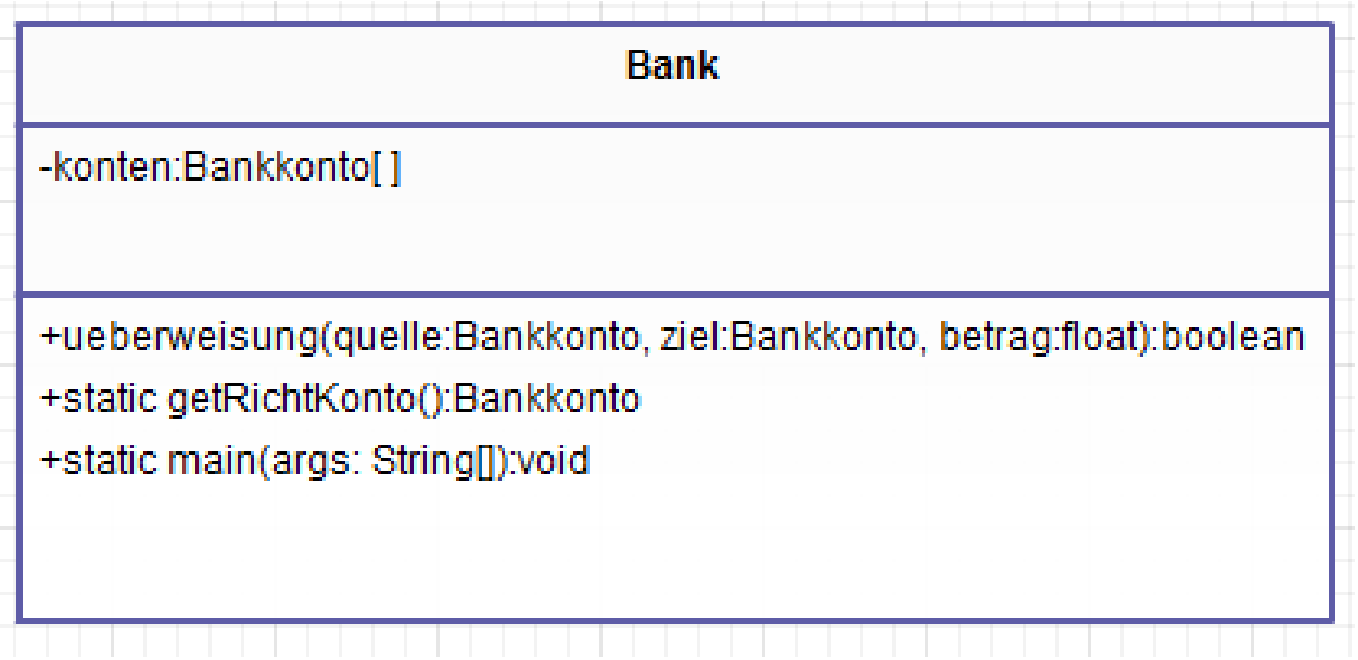
- besitzt ein Array von Bankkon-ten und die Verhaltensmethoden
- ueberweisung(quelle: Bankkonto, ziel: Bankkonto,betrag: float):boolean
  - war die überweisung erfolgreich, wird true zurückgegeben, sonst false

Erstellen Sie einer public static void main Methode der Klasse Bank eine (1) Instanz von Bank und mehrere Instanzen von Bankkonto.
Probieren Sie als "Bankräuber" direkt auf die Variable Kontostand eines Bankkontos zuzugreifen (das sollte nicht funktionieren, da die Variable private ist)- Füllen Sie die Konten und probieren Sie mehrere Überweisungen aus, auch solche, die wegen Unterdeckung abgelehnt werden sollten.

Task Bankkonto

weitere Methode der Klasse Bank

- Erstellen Sie eine statische Methode der Klasse Bank, mit der die europäische Zentralbank Konten aus dem Nichts erschaffen kann gefüllt mit 1Mio. €

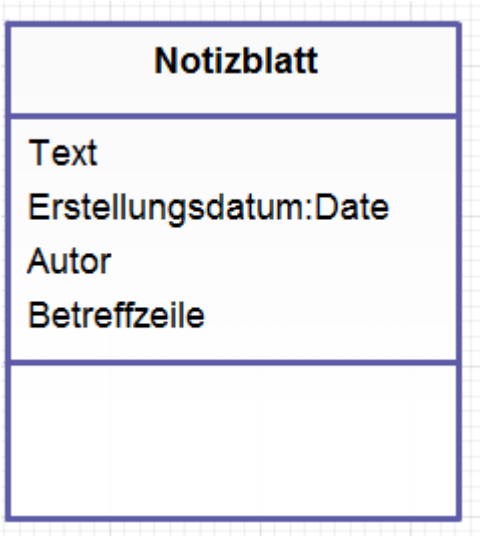


Task Notizblock

Ein Notizblock speichert Notizblätter

Ein Notizblatt speichert

- einen Text
- das Erstellungsdatum
- den Autor und
- eine Betreffzeile



Task Notizbock

Methoden der Klasse Notizblatt

- der erste Konstruktor besitzt keinen Parameter, als Betreffzeile und als Autor wird ein Standardwert eingetragen, der in der Klasse Notizblock hinterlegt ist, das aktuelle Datum wird verwendet
  - ein Datumobjekt wird erstellt mit Date date=new Date(System.currentTimeMillis());
- der zweite Konstruktur besitzt explizit die Parameter Autor und Betreffzeile, diese Angaben werden zusammen mit dem automatisch erstellten Datum im Notizblatt abgespeichert

Task Notizblock

Methoden der Klasse Notizblatt

- einen Text im Notizblatt abspeichern
- Notizblatt-Text, der Autor, die Betreffzeile und das Datum als String ausgegeben
  - Strings können durch den Additions-Operator zusammengefügt werden