ÜBUNG FÜR DIE PRAKTISCHE LEISTUNGSÜBERPRÜFUNG WS2022/2023

KLASSEN IMPLEMENTIEREN MIT BLUEJ IN JAVA LAUT KLASSENDIAGRAMM

0) Fügen Sie Ihren Namen als Kommentar oben im Sourcecode ein.

1) Implementieren Sie die Klasse Bankomat

1.1) Attribute

ort: der Standort des Bankomaten
geld: das Geld im Bankomat in EUR
karteGesteckt: eine Debit-Karte kann im Automaten gesteckt (true) oder nicht gesteckt sein (false) sein

Bankomat
- ort: String = "Spengergasse" - geld: double = 5000.0 - karteGesteckt: boolean = true
+ Bankomat() + Bankomat (String, double, boolean) + alle set-Methoden + alle get-Methoden + auszahlen(double): boolean + toString(): String + print(): void

1.2) Konstruktoren

o) Bankomat(): parameterloser Konstruktor, erzeugt ein neues Objekt und setzt die Attribute entsprechend der Default-Werte aus dem Klassendiagramm

o) Bankomat(ort: String, geld: double, karteGesteckt: boolean): erzeugt ein neues Objekt und setzt die Attribute entsprechend der übergebenen Parameterwerte

1.3) set-Methoden

Implementieren Sie die set-Methoden laut UML-Klassendiagramm mit den folgenden Plausibilitätsprüfungen:
Der Attribut-Wert von ort darf nicht null sein.
Der Attribut-Wert von geld muss größer als 0.0 sein und darf maximal 8000.0 betragen.

Wenn ein übergebener Parameter-Wert nicht gültig ist, dann muss die Ausgabe einer Meldung auf der Systemkonsole erfolgen. Beispiele:

Ausgabe auf der Konsole bei setOrt(null):
```
Fehler bei setOrt(): null-Referenz erhalten
```

Ausgabe auf der Konsole bei setGeld(-1.0):
```
Fehler bei setGeld(): ungueltigen Wert erhalten (-1.0)
```

1.4) get-Methoden

Implementieren Sie die get-Methoden laut UML-Klassendiagramm.

1.5) auszahlen(betrag: double): boolean

Das Auszahlen ist möglich, wenn der übergebene Betrag gültig ist und eine Debit-Karte im Bankomat gesteckt ist und ausreichend Geld im Bankomat vorhanden ist. Dann wird das Geld im Bankomat um den übergebenen Betrag reduziert, die Debit-Karte wird ausgegeben und es wird der Wert true von der Methode zurück geliefert.

Wenn keine Debit-Karte gesteckt ist, dann liefert die Methode false zurück (ohne Fehlermeldung).

Wenn der Attribut-Wert von geld 0, dann liefert die Methode false zurück (ohne Fehlermeldung).

Wenn nicht mehr ausreichend Geld im Bankomat vorhanden ist zum vollständigen Auszahlen des übergebenen Betrags, dann liefert die Methode false zurück (ohne Fehlermeldung).

Implementieren Sie die folgende Plausibilitätsprüfung:
Der Parameter-Wert von betrag muss größer als 0.0 sein und darf maximal 400.0 betragen, sonst liefert die Methode false zurück und es wird eine Fehlermeldung auf der Systemkonsole ausgegeben, Beispiel:

```
Fehler bei auszahlen(): ungueltigen betrag erhalten (401.0)
```

1.6) Methode toString(): String

Implementieren Sie die Methode zur Rückgabe der aktuellen Attributwerte als String wie folgt:

```
Bankomat "Spengergasse"
Geld: 5000.0 EUR
Karte gesteckt
```

```
Bankomat "Oper"
Geld: 1000.0 EUR
keine Karte gesteckt
```

Das Layout soll genau den obigen Beispielen entsprechen: es soll dreizeilig sein mit dem Wert von Ort unter Anführungszeichen, nach dem Wert von Geld soll der Text EUR angezeigt werden. Ist der Wert von karteGesteckt true, dann soll der Text „Karte gesteckt“ verwendet werden, sonst der Text „keine Karte gesteckt“

1.7) Methode print(): void

Implementieren Sie die Methode zur Ausgabe der aktuellen Attributwerte auf der Konsole.

2) BONUS: Implementieren Sie die Klasse TestBankomat

Es sind KEINE Attribute und KEINE Konstruktoren zu implementieren.
Es sind KEINE set-Methoden und KEINE get-Methoden zu implementieren.

TestBankomat
+ testToString(): void + testAuszahlen(): void

2.1) Methode testToString(): void

Erzeugt ein Testobjekt vom Typ Bankomat und gibt den Rückgabe-Wert der Methode toString() dieses Testobjekts auf der Systemkonsole aus.

2.2) Methode testAuszahlen(): void

Erzeugt ein Testobjekt vom Typ Bankomat.
Implementieren Sie einen Testfall, bei dem beim Testobjekt eine Auszahlung durchgeführt wird.
Implementieren Sie einen weiteren Testfall, bei dem beim Testobjekt keine Auszahlung durchgeführt wird, weil der Bankomat nicht ausreichend Geld dafür hat.

3) Implementieren Sie die Klasse Schleifen

Es sind KEINE Attribute und KEINE Konstruktoren zu implementieren.
Es sind KEINE set-Methoden und KEINE get-Methoden zu implementieren.

Schleifen
+ zaehleBis0(int): void + zeichneParallelogramm(int): void

3.1) Methode zahlenBis0(zahl: int): void

Beginnend mit der Zahl des übergebenen Parameter-Werts gibt die Methode alle Zahlen hinunterzählend bis zur Zahl 0 auf der Systemkonsole in einer Zeile aus. Die einzelnen Zahlen werden mit Strichpunkt getrennt. Nach der letzten Zahl 0 darf kein Strichpunkt folgen. Beispiel für zaehleBis0(10):

```
10;9;8;7;6;5;4;3;2;1;0
```

Die Plausibilitätsprüfung für den Parameter-Wert zahl darf ausnahmsweise entfallen.

3.2) Methode zeichneParallelogramm(groesse: int): void

Die Methode gibt eine Parallelogramm mit der übergebenen Größe auf der Systemkonsole aus.

```
* * * *
  * * * *
    * * * *
      * * * *
```

Die Plausibilitätsprüfung für den Parameter-Wert groesse darf ausnahmsweise entfallen.

GUTES GELINGEN UND VIEL SPASS!