

Programmieren und Software-Engineering II

Übung 16

Name: _____ Klasse: _____ Datum: _____

Aufgabe 1: KFZ

Bei der Berechnung der **jährlichen Versicherungsprämie** von Kraftfahrzeugen (Kfz) unterscheidet man zwischen PKWs, LKWs und Tank-LKWs. Jedes **Kfz** hat eine Marke und einen Typ (z.B. Marke: „Skoda“, Typ: „Oktavia“)

Bei **PKWs** wird zur Berechnung der jährlichen Versicherung die Leistung (in PS) herangezogen. Die Versicherungssumme beträgt $200 + 7 \cdot \text{Leistung}$.

Bei **LKWs** hängt die Versicherung vom Gewicht des Fahrzeugs ab. Sie beträgt $\text{Gewicht} \cdot 0,7$.

Bei **Tank-LKWs** wird zusätzlich die Gefahrenstufe bei der Berechnung der Versicherungssumme berücksichtigt (1 für Lebensmittel, 5 für Transporte hochgiftiger Chemikalien). Die für normale LKWs berechnete Prämie wird mit diesem Wert multipliziert.

Beispiel:

PKW mit 90 PS	Vers.Prämie = $200 + 7 \cdot 90 = 830$ Euro
LKW mit 3000 kg Gewicht	Vers.Prämie = $3000 \cdot 0,7 = 2100$ Euro
Tank-LKW mit 2000 kg Eigengewicht, Gefahrenstufe 2	Vers.Prämie = $2000 \cdot 0,7 \cdot 2 = 2800$ Euro

Ein Fuhrpark wiederum enthält eine gewisse Menge an Fahrzeugen (maximal 10).

Folgende Aufgaben sind zu realisieren:

- Erstelle ein UML-Diagramm für den beschriebenen Sachverhalt.
- Verschiedene Kraftfahrzeuge werden in einem **Array** gespeichert.
- Dann werden alle Fahrzeuge **aufsteigend sortiert** nach Marke, Typ und Versicherungsprämie ausgegeben.
- Anschließend wird das Fahrzeug mit der **geringsten Versicherungssumme** bzw. der **LKW** mit der **höchsten Versicherungsprämie** ausgegeben.
- Zum Schluss wird erneut eine **Liste** ausgegeben. **Zuerst** alle Pkws, dann alle LKWs und abschließend alle Tank-LKWs. Innerhalb dieser **Kategorien** wird aufsteigend nach der Versicherungsprämie sortiert. Nach jeder Kategorie wird die jeweilige Gesamtsumme angedruckt (Verwende entweder instanceof bzw. die Methode getClass.getSimpleName()).
- Füge einzelne Fahrzeuge zu einem **Fuhrpark** hinzu. Wie würden in diesem Fall die obigen Methoden aussehen?

Das **Hauptprogramm** besitzt ca. folgenden **Aufbau**:

```
Kfz[] ar = new Kfz[8];
ar[0]=new Pkw("Skoda","Oktavia",90);
ar[1]=new Lkw("Volva","Truck",8000);
ar[2]=new Pkw("Skoda","Fabia",70);
ar[3]=new TankLkw("Man","Dangertruck",7500,2);
ar[4]=new Lkw("Mercedes","Monstertruck",7000);
ar[5]=new Pkw("Skoda","Oktavia",120);
ar[6]=new Lkw("Man","Truck",9000);
ar[7]=new Pkw("Skoda","Oktavia",75);

// Ausgabe unsortiert
System.out.println("Unsortiert : ");
printAll(ar);

// Sortieren und ausgeben
System.out.println("\nSortiert : ");
sort(ar);
printAll(ar);

// cheapest bzw. mostExpensive
System.out.println("\nBillig : ");
Kfz cheap;
cheap=getcheapest(ar);
if(cheap!=null){
    cheap.print();           // oder mit toString()
}else{
    System.out.println("no Kfz");
}

System.out.println("\nTeuer : ");
Lkw expensive;
expensive=getMostExpensiv(ar);
if(expensive!=null){
    expensive.print();       // oder mit toString()
}....

// Sortieren nach Kategorie und ausgeben
System.out.println("\nSortiert : ");
sortByCategory(ar);
printAll(ar);

// Fuhrpark anlegen und Fahrzeuge hinzufügen
...
```

Programmieren und Software-Engineering II

Übung 16

Aufgabe 2: Spargelbauer

Spargelbauer „Langer Jan“ braucht dringend eine einfache Personalverwaltung. Dabei sind folgende Gegebenheiten zu berücksichtigen:

- In dem Betrieb arbeiten sowohl Angestellte (vorwiegend fix angestelltes Büropersonal) als auch Arbeiter (z.B. tagesbeschäftigte Erntehelfer).
- Für alle im Betrieb befindlichen Personen müssen ID, Name und Sozialversicherungsnummer gespeichert werden.
- Ein Angestellter speichert zusätzlich seinen Grundgehalt und eine Überstundenpauschale.
- Ein Arbeiter speichert zusätzlich Stundenlohn und aktuelles Stundenkonto (=Anzahl der gearbeiteten Stunden).

Für die einzelnen Variablen gelten dabei folgende Informationen:

- Id, String: wird beim Anlegen automatisch aus den ersten beiden Buchstaben des Namens und einer zufälligen Zahl zwischen 1000 und 9999 gebildet.
- Name, String: keine Einschränkungen, Standardwert „unknown“
- Sozialversicherungsnummer, int: ganze positive Zahl zwischen 1000 und 9999, Standardwert 1000
- Grundgehalt, double: muss zwischen 1600 und 10000 liegen, Standardwert 2000
- Überstundenpauschale, double: Prozentwert zwischen 0 und 100, Standardwert 5
- Stundenlohn, double: muss zwischen 13,50 und 39,90 liegen, Standardwert 22,22
- Stundenkonto, int: gibt die Anzahl der bisher geleisteten Stunden an und muss zwischen 0 und 200 liegen, Standardwert 160

Aufgabenstellung:

- Erstelle die Klassen Person, Employee(=Angestellter) und Worker(=Arbeiter) in einer einfachen Vererbungshierarchie. Lege alle benötigten Setter und Getter an.
- Erstelle für die Klassen Employee und Worker je eine Methode „computeMoney()“, die jeweils den zu bezahlenden Betrag berechnet. Für den Worker berechnet sich der Betrag aus $\text{Stundenlohn} \times \text{Stundenkonto}$, für den Employee aus $\text{Grundgehalt} \times (1 + \text{Überstundenpauschale} / 100)$.
- Erstelle im Hauptprogramm je ein 5stelliges Array von Arbeitern und Angestellten und berechne, welche Gruppe dem „Langer Jan“ mehr Geld kostet.