

Versuchen Sie immer, Ihren Code zu kommentieren!

Erstellen Sie für jede Aufgabe ein Klassendiagramm, bevor Sie mit der Aufgabe beginnen.

Aufgabe 1

Erstellen Sie für die Familie der Simpsons 5 verschiedene Klassen und verwenden Sie Vererbung (d.h. eine Basisklasse) um möglichst keine Informationen doppelt zu speichern.

Homer

Eigenschaften:

Vorname: Homer
Nachname: Simpson
Hautfarbe: Gelb
Wohnort: Evergreen Terrace 742, Springfield
Zähler: Anzahl gegessener Donuts

Methoden:

- Sich Vorstellen (Mit Name, Vorname, Wohnort und Hautfarbe)
- Donuts essen (erhöht Anzahl um Eins)
- Zähler ausgeben

Marge

Eigenschaften:

Vorname: Marge
Nachname: Simpson
Hautfarbe: Gelb
Wohnort: Evergreen Terrace 742, Springfield
Zähler: Wie oft um den Haushalt gekümmert

Methoden:

- Sich Vorstellen (Mit Name, Vorname, Wohnort und Hautfarbe)
- Um den Haushalt kümmern (erhöht Anzahl um Eins)
- Zähler ausgeben

Bart

Eigenschaften:

Vorname: Bart
Nachname: Simpson
Hautfarbe: Gelb
Wohnort: Evergreen Terrace 742, Springfield
Zähler: Wie oft Skateboard gefahren

Methoden:

- Sich Vorstellen (Mit Name, Vorname, Wohnort und Hautfarbe)
- Skateboard fahren (erhöht Anzahl um Eins)
- Zähler ausgeben

Lisa

Eigenschaften:

Vorname: Lisa

Nachname: Simpson

Hautfarbe: Gelb

Wohnort: Evergreen Terrace 742, Springfield

Zähler: Wie oft Saxophon gespielt

Methoden:

- Sich Vorstellen (Mit Name, Vorname, Wohnort und Hautfarbe)
- Saxophon spielen (erhöht Anzahl um Eins)
- Zähler ausgeben

Maggie

Eigenschaften:

Vorname: Maggie

Nachname: Simpson

Hautfarbe: Gelb

Wohnort: Evergreen Terrace 742, Springfield

Zähler: Wie oft am Schnuller genuckelt

Methoden:

- Sich Vorstellen (Mit Name, Vorname, Wohnort und Hautfarbe)
- Am Schnuller nuckeln (erhöht Anzahl um Eins)
- Zähler ausgeben

Aufgabe 2

Sie entwickeln für eine Firma eine Software zur Gehaltsberechnung in der Personalabteilung.

Es gibt drei Arten von Mitarbeitern:

- Angestellte
- Externe Mitarbeiter
- Praktikanten

Für jeden Mitarbeiter werden der Name, Vorname und die zur Gehaltsberechnung notwendigen Daten gespeichert.

Angestellte werden nach Tarifgruppen und Alter bezahlt. Es gibt die folgenden vier Tarifgruppen:

Tarifgruppe	Gehalt
A	2560 €
B	3000 €
C	3200 €
D	5400 €

Die Formel für das Gehalt eines Angestellten lautet:

$$\text{Gehalt} = \text{Tarifgruppe} * \left(1 + \frac{\text{Alter} - 25}{100} \right)$$

Externe Mitarbeiter werden nach geleisteten Projektstunden bezahlt. Für jede Projektstunde wird ein Betrag von 75 € bezahlt.

Die Praktikanten bekommen eine monatliche Vergütung, je nach Abteilung in der diese tätig sind. Es gibt drei Abteilungen: Entwicklung, Vertrieb und Produktion. Ein Praktikant aus dem Vertrieb bekommt monatlich 820 €, ein Praktikant aus der Produktion 710 € und einer aus der Entwicklung 935 €.

Legen Sie einige Mitarbeiter mit entsprechenden Werten an und geben Sie die Gehälter aus.

Aufgabe 3

Modellieren Sie Konten, Girokonten, Sparkonten und Festgeldkonten. Ein Konto hat genau einen Inhaber, d.h. eine Person mit sinnvollen Attributen, eine Kontonummer, sowie einen Kontostand. Zudem wird das Anlegedatum des Kontos gespeichert. Für die Zinsberechnung hat jedes Konto einen Zinssatz namens Habenzinssatz. Es sollen die Methoden `Einzahlen()`, `Auszahlen()`, `BerechneZinsen()` und `KontostandAusgeben()` enthalten sein.

Ein Girokonto ist ein Konto, das einen geduldeten Überziehungsrahmen (Dispokredit) besitzt. Es wird mit 0 % verzinst. Wird das Konto überzogen ist ein Überziehungssatz (Sollzinsen) von 16% fällig. Außerdem verfügt ein Girokonto über eine Kategorie, welcher einen Gebührentarif für das Konto angeben soll. Kategorien sind Standard, Schüler, Student und Gewerblich.

Ein Sparkonto ist ein Konto, welches ein Guthabenzinssatz von 0,5 % besitzt.

Ein Festgeldkonto hat eine Mindestlaufzeit. Es wird mit 2% verzinst. Zudem soll eine Methode geben, welche die Restlaufzeit des Kontos ausgibt. Verwenden je nach Restlaufzeit sinnvolle Zeitangaben wie Jahre, Monate oder Wochen.

Legen Sie verschiedene Konten an und testen Sie alle Methoden der Konten.

Aufgabe 4

Es gibt 11 Spieler, mit den Eigenschaften `name`, `alter`, `stärke`, `torschuss`, `motivation` und `tore`. `Stärke`, `Torschuss`, `Motivation` sind Werte von 1 (schlecht) bis 10 (super). Bei den Spielern gibt es noch einen Sonderling, den Torwart. Als zusätzliche Eigenschaft hat er `reaktion`, damit wird später entschieden, ob er die Torschüsse hält oder nicht. Der Torwart ist ein Spezialfall eines Spielers.

Der Spieler erhält zwei Methoden `AddTor()` und `SchiesstAufTor()`. Die Methode `AddTor()` erhöht die Anzahl der Tore, wenn der Spieler ein Tor geschossen hat. Die Methode `SchiesstAufTor()` ermittelt die Torschussqualität. Es wird eine zufällige Zahl im Bereich von 1-10 gewählt, die abhängig von der Torschussqualität des Spielers ist.

Verwenden Sie für die Methode SchiesstAufTor() die Implementierung aus der beiliegenden Datei.

Der Torwart erhält eine Methode HältDenSchuss(), welcher die Torschussqualität eines Spielers übergeben wird. Die Methode entscheidet, ob der abgegebene Torschuss pariert oder durchgelassen wird (ebenfalls mit einem zufälligen Ausgang).

Verwenden Sie für die Methode HältDenSchuss() die Implementierung aus der beiliegenden Datei.

Neben den Spielern haben wir einen Trainer mit den Eigenschaften name, alter und erfahrung.

Alle Objekte haben Methoden zum Lesen ihrer eigenen Attribute (z.B. GetName, GetAlter, GetTore, GetMotivation, GetStärke, GetReaktion, GetErfahrung, usw.)

Die Konstruktoren sind wie folgt aufgebaut:

```
public Spieler(string name, int alter, int stärke, int torschuss,
int motivation)
```

```
public Torwart(string name, int alter, int stärke, int torschuss,
int motivation, int reaktion)
```

```
public Trainer(string name, int alter, int erfahrung)
```

Eine Mannschaft ist eine Klasse mit den Eigenschaften name, trainer, torwart und spieler (als Array). Es gibt wieder einen Konstruktor und die Get- Methoden für die Eigenschaften. Zusätzlich besitzt die Klasse Mannschaft die Methoden GetStärke() und GetMotivation(), die die durchschnittliche Stärke, bzw. Motivation der Mannschaft als Zahlenwert wiedergibt. D.h. Die Methoden addieren die Werte der Spieler zu einer Summe auf und liefern den Mittelwert zurück.

Zuletzt gibt es eine Methode GetKader() die wie folgt implementiert wird:

```
public Spieler[] GetKader() {
    return spieler;
}
```

Ein Spiel zwischen zwei Mannschaften hat folgende Attribute: nameHeimMannschaft, nameGastMannschaft, punkteHeim, punkteGast. Zudem fünf Get-Methoden :

```
string GetHeimMannschaft()
string GetGastMannschaft()
int GetHeimPunkte()
int GetGastPunkte()
string GetErgebnisText()
```

Die Methode GetErgebnisText() gibt die Namen und die Punkte aus.

Die Methode StarteSpiel() finden Sie in der beiliegenden Datei.

Nachdem Sie Klassen erzeugt haben, legen Sie im Hauptprogramm zwei Mannschaften wie folgt an (vollständig in der Textdatei):

```
// Mannschaft 1
Trainer trainer1 = new Trainer("Juergen Klinsmann", 34, 9);
Torwart torwart1 = new Torwart("J. Lehmann", 36, 8, 1, 9, 7);
Spieler[] spieler1 = new Spieler[10];
spieler1[0] = new Spieler("P. Lahm", 23, 9, 5, 9);
...

Mannschaft m1 = new Mannschaft("Deutschland WM 2006", trainer1,
torwart1, spieler1);
// Mannschaft 2
Trainer trainer2 = new Trainer("Carlos Alberto Parreira", 50, 3);
Torwart torwart2 = new Torwart("Dida", 25, 9, 1, 6, 8);
Spieler[] spieler2 = new Spieler[10];
spieler2[0] = new Spieler("Cafu", 33, 8, 4, 6);
...

Mannschaft m2 = new Mannschaft("Brasilien WM 2006", trainer2,
torwart2, spieler2);
```

Starten Sie anschließend ein Spiel zwischen diesen Mannschaften wie folgt (kann ebenfalls der Textdatei entnommen werden):

```
Spiel spiel = new Spiel();

Console.WriteLine("-----");
Console.WriteLine("Start des Spiels zwischen");
Console.WriteLine();
Console.WriteLine(m1.GetName());
Console.WriteLine(" Trainer: " + m1.GetTrainer().GetName());
Console.WriteLine();
Console.WriteLine(" und");
Console.WriteLine();
Console.WriteLine(m2.GetName());
Console.WriteLine(" Trainer: " + m2.GetTrainer().GetName());
Console.WriteLine("-----");
spiel.StarteSpiel(m1, m2);
Console.WriteLine();
Console.WriteLine("-----");
Console.WriteLine(spiel.GetErgebnisText());
Console.WriteLine("-----");
```