

Software Engineering

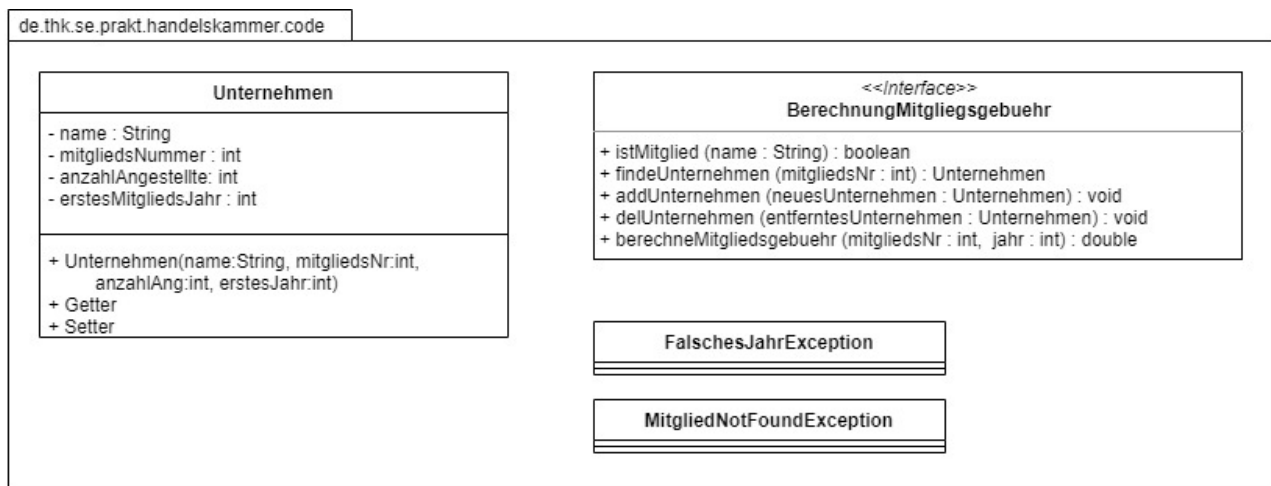
Praktikumsversuch 2, Gruppe E - Hausaufgaben -

Ziel: Erstellung von JUnit-Testfällen

Abgabe der Lösungen: Bis zum 14.12., 08:00 Uhr morgens, im Master-Branch Ihres Gitlab-Repositories P2SE<IhreTeilnehmernummer> (z.B. <https://gitlab.nt.fh-koeln.de/gitlab/se/SE29/P2SE29.git> für Teilnehmer 29). Abzugeben ist das vollständige IntelliJ-Projekt.

In Ihrem Gitlab-Repository zu diesem Praktikumsversuch habe ich Ihnen ein IntelliJ-Projekt zur Verfügung gestellt, welches Sie als Grundlage Ihrer Implementierung verwenden müssen.

Es soll für eine Handelskammer ein kleines Programm zur Berechnung der Mitgliedsgebühren für die Unternehmen in dieser Handelskammer erstellt werden. Es existiert hierfür das folgende Klassendiagramm:



Ein Unternehmen hat seinen Namen, seine eindeutige Mitgliedsnummer, seine Angestelltenanzahl und das erste Jahr seiner Mitgliedschaft als Attribute.

Bereits implementiert sind die Exception-Klassen, die Klasse `Unternehmen` und das Interface `BerechnungMitgliedsgebuehr`.

Ihre Aufgaben:

H 2.1 Implementierung der Schnittstellen-Klasse

Erstellen Sie eine Implementierung der Interface-Klasse `BerechnungMitgliedsgebuehr`.

Diese Implementierung soll eine private `ArrayList` mit den Mitgliedsunternehmen besitzen (im Folgenden als Unternehmensliste bezeichnet).

Die Semantik der Methoden des Interface ist:

- Die Methode `istMitglied` nimmt den Namen eines Unternehmens als Parameter und liefert `true`, falls ein Unternehmen mit diesem Namen in der Unternehmensliste existiert und `false`, falls kein Unternehmen mit diesem Namen in der Unternehmensliste existiert.
- Die Methode `findeUnternehmen` nimmt eine Mitgliedsnummer als Parameter und liefert das Unternehmen aus der Unternehmensliste mit dieser Mitgliedsnummer als Ergebnis. Sie wirft die Exception `MitgliedNotFoundException`, falls sich kein Unternehmen mit dieser Mitgliedsnummer in der Unternehmensliste befindet.
- Die Methode `addUnternehmen` fügt ein Unternehmen zur Unternehmensliste hinzu.
- Die Methode `delUnternehmen` entfernt ein Unternehmen aus der Unternehmensliste.
- Die Methode `berechneMitgliedsgebuehr` nimmt eine Mitgliedsnummer und ein Jahr als Parameter und berechnet die Mitgliedsgebühr für dieses Unternehmen für das übergebene Jahr:
 - Bei einem Unternehmen mit einem ersten Mitgliedsjahr (= Attribut `erstesMitgliedsJahr`) welches exakt dem übergebenen Jahr entspricht, beträgt die Mitgliedsgebühr 0,10 € pro Mitarbeiter,
 - Falls das erste Mitgliedsjahr des Unternehmens 1 oder 2 Jahre vor dem übergebenen Jahr liegt, beträgt die Mitgliedsgebühr 0,09 € pro Mitarbeiter,
 - Falls das erste Mitgliedsjahr des Unternehmens 3 oder 4 Jahre vor dem übergebenen Jahr liegt, beträgt die Mitgliedsgebühr 0,08 € pro Mitarbeiter,
 - Falls das erste Mitgliedsjahr des Unternehmens 5 oder 6 Jahre vor dem übergebenen Jahr liegt, beträgt die Mitgliedsgebühr 0,07 € pro Mitarbeiter,
 - Falls das erste Mitgliedsjahr des Unternehmens 7 oder mehr Jahre vor dem übergebenen Jahr liegt, beträgt die Mitgliedsgebühr 0,05 € pro Mitarbeiter.
- Die Methode `berechneMitgliedsgebuehr` wirft die Exception `FalschesJahrException`, falls das übergebene Jahr echt kleiner als das erste Mitgliedsjahr des Unternehmens ist.
- Die Methode `berechneMitgliedsgebuehr` reicht die geworfene Exception `UnternehmenNotFoundException` an den Aufrufer weiter.

H 2.2 Realisierung von JUnit-Testfällen

Erstellen Sie im Paket `de.thk.se.prakt.handelskammer.test` JUnit-Testklassen zur Prüfung der folgenden Eigenschaften Ihrer Implementierung:

- (1) Wenn ein Unternehmen mit dem Namen „Microsoft“ in der Unternehmensliste vorhanden ist, liefert der Methodenaufruf `istMitglied („Microsoft“)` `true`.
- (2) Wenn ein Unternehmen mit dem Namen „Microsoft“ in der Unternehmensliste nicht vorhanden ist, liefert der Methodenaufruf `istMitglied („Microsoft“)` `false`.
- (3) Wenn kein Unternehmen mit der Mitgliedsnummer 531 in der nicht-leeren Unternehmensliste existiert, wirft der Methodenaufruf `findeUnternehmen(531)` die Exception `MitgliedNotFoundException`.

- (4) Wenn das Unternehmen mit der Mitgliedsnummer 531 in der nicht-leeren Unternehmensliste existiert, liefert der Methodenaufruf `findeUnternehmen(531)` genau dieses Unternehmen.
- (5) Wenn die Unternehmensliste leer ist, wirft der Methodenaufruf `findeUnternehmen(531)` die Exception `MitgliedNotFoundException`.
- (6) Für ein Unternehmen mit der Mitgliedsnummer 2573 und dem ersten Mitgliedsjahr 2003 wirft der Methodenaufruf `berechneMitgliedsgebuehr(2573, 2000)` die Exception `FalschesJahrException`.
- (7) Implementieren Sie parametrisierte Testfälle für die folgenden Prüfungen:
- Für ein in der Unternehmensliste existierendes Unternehmen mit Mitgliedsnummer 96, mit dem ersten Mitgliedsjahr 2016 und der Angestelltenanzahl 57 liefert der Aufruf von `berechneMitgliedsgebuehr(96, 2016)` das korrekte Ergebnis.
 - Für ein in der Unternehmensliste existierendes Unternehmen mit Mitgliedsnummer 321, mit dem ersten Mitgliedsjahr 2020 und der Angestelltenanzahl 1346 liefert der Aufruf von `berechneMitgliedsgebuehr(321, 2021)` das korrekte Ergebnis.
 - Für ein in der Unternehmensliste existierendes Unternehmen mit Mitgliedsnummer 135, mit dem ersten Mitgliedsjahr 2016 und der Angestelltenanzahl 180 liefert der Aufruf von `berechneMitgliedsgebuehr(135, 2020)` das korrekte Ergebnis.
 - Für ein in der Unternehmensliste existierendes Unternehmen mit Mitgliedsnummer 1, mit dem ersten Mitgliedsjahr 2010 und der Angestelltenanzahl 256 liefert der Aufruf von `berechneMitgliedsgebuehr(1, 2015)` das korrekte Ergebnis.
 - Für ein in der Unternehmensliste existierendes Unternehmen mit Mitgliedsnummer 4567, mit dem ersten Mitgliedsjahr 1999 und der Angestelltenanzahl 739 liefert der Aufruf von `berechneMitgliedsgebuehr(4567, 2021)` das korrekte Ergebnis.
 - Das jeweils korrekte Ergebnis müssen Sie hierbei selbst ausrechnen.
- (8) Erstellen Sie im Paket `de.thk.se.prakt.handelskammer.test` eine JUnit-Test-Suite, durch die alle erstellten Testklassen aufgerufen werden.