Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

Entwicklung komplexer Software-Systeme

Praktikumsblatt 1
Gruppe C
- Hausaufgaben -

Ziel: Implementierung von Entwurfsmustern

Abgabe der Lösungen: Bis zum 29.11., 08:00 Uhr morgens, im Master-Branch des Gitlab-Repositories P1EKS<IhreTeamnummer>. Abzugeben ist das vollständige IntelliJ-Projekt.
Hinweis: Ihr Gitlab-Repository zu diesem Praktikumsversuch enthält ein IntelliJ-Projekt, welches Sie als Grundlage für Ihre Implementierung verwenden sollen – Sie sollen also den Ordner P1EKS< IhreTeamnummer> als IntelliJ-Projekt öffnen.

Achtung: Seien Sie sorgfältig bei der Implementierung. Es ist sehr wichtig, dass Sie die Aufgabenstellung exakt und korrekt umsetzen. Verwenden Sie keine öffentlichen Attribute.

Aufgaben:

H1.1 Entwurfsmuster „Schablonenmethode“ implementieren

Wir haben eine Tischleuchte, die in den Farben rot, gelb, grün, blau und weiß leuchten kann. Diese Leuchte soll in dieser Aufgabe als Klasse `Leuchte` implementiert werden. Diese Klasse besitzt die Methode `einschalten()` (die Schablonenmethode). In der Methode `einschalten()` soll über eine Schleife die Farbe genau 10 Mal gewechselt werden durch Aufruf der Methode `farbeWechseln()`. In welcher Art die Farbe gewechselt wird, soll aber variabel bleiben. Durch das zu implementierende Muster Schablonenmethode sollen 3 unterschiedliche Strategien zum Farbwechsel umgesetzt werden. Die drei verschiedenen Strategien sind:

- es wird auf eine zufällig ausgewählte Farbe gewechselt
- es wird immer auf eine fest voreingestellte Farbe gewechselt (d.h. die Farbe wird gar nicht gewechselt) (Bem.: die Farbe können Sie sich selbst aussuchen)
- es wird zyklisch zwischen den Farben gewechselt: rot, gelb, grün, blau, weiß, rot, gelb, grün, blau, …

Dabei soll bei jedem Aufruf der Methode `farbeWechseln()` die jeweils neue Farbe auf der Console ausgegeben werden.

a) Implementieren Sie die Leuchte mit den drei Strategien unter Anwendung des Musters Schablonenmethode.

b) Schreiben Sie ein kleines Testprogramm, welches nacheinander jede der drei Strategien anwendet.

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

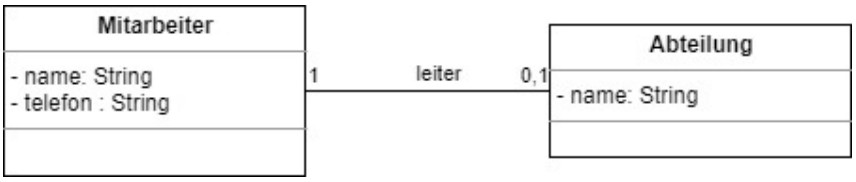
H1.2 Entwurfsmuster „Kompositum“ implementieren

Wir wollen das Kompositum-Muster nutzen, um die Organisationsstruktur unserer Firma darzustellen. In unserer Firma gibt es:

- Mitarbeiter, die einen Namen und eine Telefonnummer besitzen
- Abteilungen, die einen Namen und einen Mitarbeiter als Abteilungsleiter besitzen. Eine Abteilung kann aus weiteren (Unter-)Abteilungen und aus Mitarbeitern bestehen. Jede Abteilung soll die Anzahl der Mitarbeiter in dieser Abteilung (inklusive des Leiters und der Mitarbeiter in den Unter-Abteilungen) durch Aufruf der Methode `getMitarbeiterAnzahl()` angeben können. Diese Methode soll aber auch für einen einzelnen Mitarbeiter einen sinnvollen Wert ergeben. Die gesamte Firma soll hierbei auch als eine Abteilung dargestellt werden.

Die gesamte Hierarchie und auch die Struktur einer einzelnen Abteilung soll durch einen Methodenaufruf (`print()`) ausgedruckt werden können.

Die Klassen `Mitarbeiter` und `Abteilung` ohne die Darstellung einer Hierarchie und ohne Methoden sehen folgendermaßen aus:



Die Abteilung verweist über die Assoziation `leitet` auf den Mitarbeiter, der diese Abteilung leitet.

Die Methode `print()` soll für Mitarbeiter und Abteilungen möglich sein und eine Ausgabe der Hierarchie mit Einrückungen ermöglichen. Deshalb erhält diese Methode einen Parameter `einrueckung` vom Typ `String`, der die jeweilige Einrückung angibt:

- Auf der obersten Ebene erfolgt keine Einrückung (Parameter = `""` = leerer String).
- Auf der Eben darunter erfolgt eine Einrückung um 4 Blanks (Parameter = `"    "`).
- Auf der Ebene darunter erfolgt die Einrückung um 4 weitere Blanks,
- usw.

Für einen einzelnen Mitarbeiter soll `print()` den Namen und die Telefonnummer ausgeben (Beispiel: Schmidt, 3344).
Für eine Abteilung soll `print()` den Namen der Abteilung und den Namen und die Telefonnummer des Abteilungsleiters ausgeben und dann – mit Einrückung – die dieser Abteilung zugeordneten Mitarbeiter und Unterabteilungen.
Beispiel:

```
Abteilung Entwicklung, Leiter: Petersen, 5566
    Hansen, 6789
    Müller, 9876
    Abteilung Test, Leiter: Jensen, 5544
        Wagner, 2929
        Goethe, 2345
```

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

Ihre Aufgaben:
Realisieren Sie die Darstellung der Mitarbeiter und der Abteilungen durch Anwendung des Kompositum-Musters. Realisieren Sie hierbei auch Methoden zum Hinzufügen und Entfernen einer Struktur zu einer Abteilung (also Unter-Abteilungen und/oder Mitarbeiter).
Implementieren Sie die Methoden `print(einrueckung : String)` und `getMitarbeiterAnzahl()` entsprechend des Kompositum-Musters.

Erstellen Sie eine Klasse `Client`, die in einer `main`-Methode folgende Firmenstruktur aufbaut:

- Firma „Best Software Development“ mit Firmenchef “Chef”, Telefon = 1
- Mit folgenden Abteilungen, Unterabteilungen und Mitarbeitern:
  - „Projekte“, Leiterin „Meier“, Telefon = 3400,
    - Mit Mitarbeiterin „Kerstin“, Telefon = 1000
    - Mit Mitarbeiter „Peter“, Telefon = 1200
  - „Entwicklung“, Leiterin „Petersen“, Telefon = 5566
    - Mit Mitarbeiter „Hansen“, Telefon = 6789
    - Mit Mitarbeiterin „Müller“, Telefon = 9876
    - Mit Unterabteilung „Test“, Leiter „Jensen“, Telefon = 5544
      - Mit Mitarbeiterin „Wagner“, Telefon = 2929
      - Mit Mitarbeiter „Goethe“, Telefon = 2345
  - „Finanzen“, Leiter = „Knete“, Telefon = 3200
    - Mit Mitarbeiter „Michael“, Telefon = 2100
    - Mit Mitarbeiterin „Stefanie“, Telefon = 2200
    - Mit Unterabteilung „Buchhaltung“, Leiter „Dieter“, Telefon = 3300
      - Mit Mitarbeiterin „Sabrina“, Telefon = 4300
      - Mit Mitarbeiter „Torsten“, Telefon = 4400
    - Mit Unterabteilung „Controlling“, Leiterin „Petra“, Telefon = 5400
      - Mit Mitarbeiter „Detlef“, Telefon = 5600
      - Mit Mitarbeiterin „Silke“, Telefon = 5700

In der `main`-Methode sollen dann folgende Aufrufe erfolgen:

- `getMitarbeiterAnzahl()` für die Firma „Best Software Development“
- `print("")` für die Firma „Best Software Development“
- `getMitarbeiterAnzahl()` für die Abteilung „Entwicklung“
- `print("")` für die Abteilung „Entwicklung“
- `getMitarbeiterAnzahl()` für die Abteilung „Finanzen“
- `print("")` für die Abteilung „Finanzen“
- `getMitarbeiterAnzahl()` für die Mitarbeiterin „Silke“
- `print("")` für die Mitarbeiterin „Silke“
- `getMitarbeiterAnzahl()` für die Mitarbeiterin „Petersen“
- `print("")` für die Mitarbeiterin „Petersen“