Herr Weiß, der der Projektleiter für individuelle Kundensoftware, möchte, dass Sie ihn bei der Durchführung eines neuen Projekt-Auftrags unterstützen. Die „Aix-Car-Group Ltd.“, war so angetan von der individuellen Software für Mietwagen, dass Sie darum gebeten hat, eine Software für die Verwaltung von Gehältern ihrer unterschiedlichen Mitarbeiter zu unterstützen.

Schon wieder waren Sie, bedingt durch den Berufsschulunterricht, weder bei den Gesprächen mit dem Kunden noch bei den ersten Design-Entwürfen dabei. Herr Weiß übergibt Ihnen das Gesprächsprotokoll und den ersten Entwurf Ihrer Fachkollegen und bittet Sie sich mit den Anforderungen vertraut zu machen.

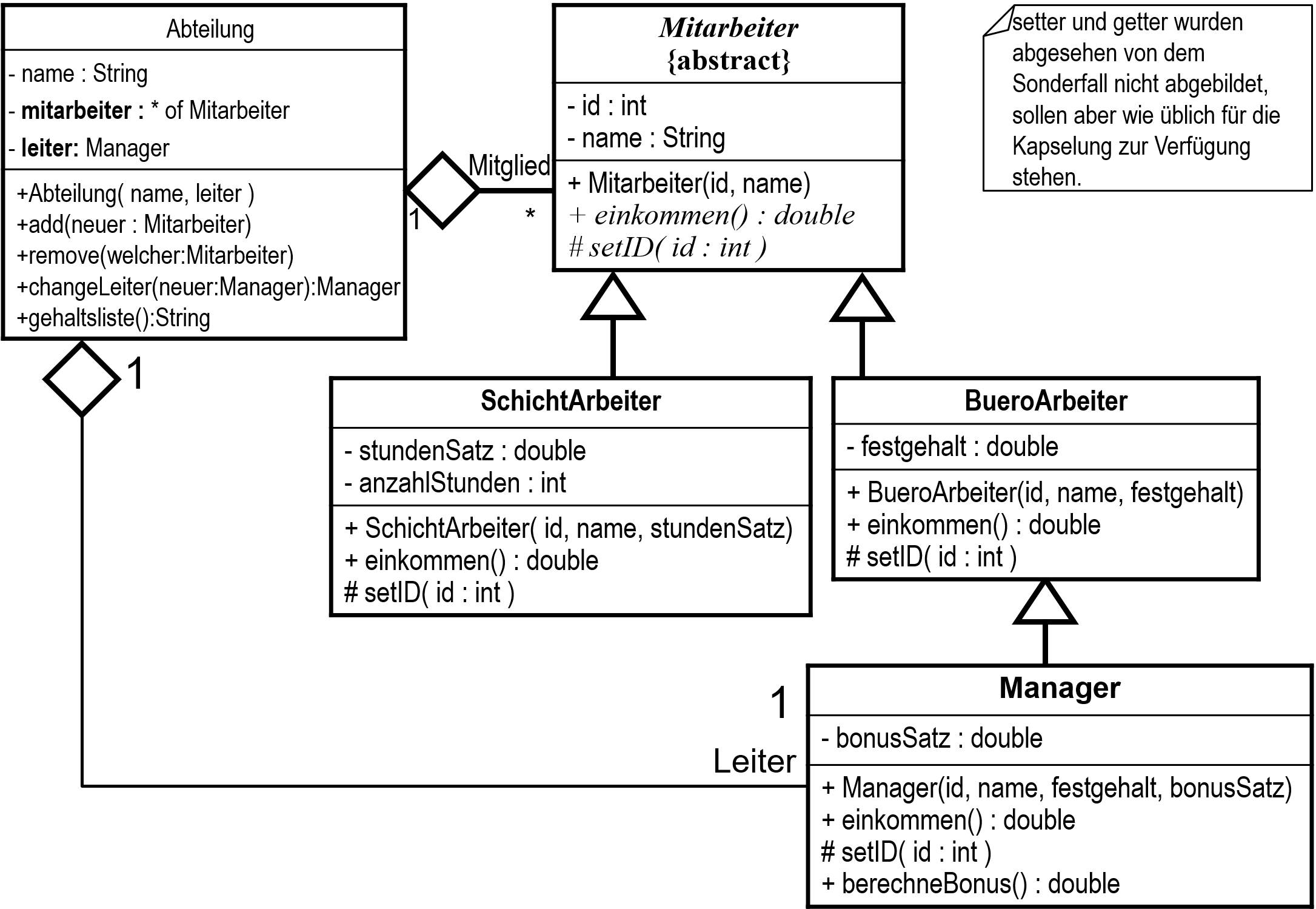
Gesprächsprotokoll des Kundengesprächs

Anwesend: Herr Weiß und Frau Kallenbring

* Die Software soll je nach Mitarbeitertyp die Gehälter unterschiedlich bestimmen können. Schichtarbeiter werden nach Stunden und Stundensatz bezahlt. Büroarbeiter eralten ein Festgehalt. Manager erhalten neben einem Festgehalt auch einen Bonus, abhängig von einem individuell vereinbarten Prozentsatz.
* IDs werden einmalig erstellt und unterliegen bestimmten Regeln:
  + Mitarbeiter IDs sind immer positiv und vierstellig.
  + Schichtarbeiter IDs sind positiv, vierstellig und beginnen mit der Ziffer 3.
  + Bueroarbeiter IDs sind positiv, vierstellig und beginnen mit 5.
  + Manager IDs sind positiv, vierstellig und beginnen mit 50.
* Eine Verwaltung der Mitarbeiter soll in der Lage sein,
  + Eine Gehaltsliste mit abschließender Gehaltssumme zu erstellen.
  + die aktuellen Mitarbeiterdaten zu speichern und wieder herzustellen.
  + Neue Mitarbeiter einzustellen und
  + Miarbeiter zu entlassen.

Ergebnis des Design-Meetings vom 11.09.2021

Anwesend: Herr Weiß, Frau Kläwer, Frau Schwaygsam und Herr Grünohr



Uff!!!

Diese Rechtecke kommen mir noch vage bekannt vor, aber da sind so viele neue Symbole...



Herr Weiß hat Verständnis für Ihre Panik und will Sie unterstützen. Der erste Tipp den er gibt ist "Keine Panik. Viele kleine Schritte führen zum Ziel." Er drückt Ihnen eine Liste in die Hand, anhand der Sie sich orientieren können.

**Aufgabe 1**

**1. Erstmal mach Dir das mit den +, -, : und den Unterschied zwischen () und keine Klammer wieder vertraut.**

**2. Fange dann mit den Methoden an, die so heißen wie die Klassen.**

**3. Dann schau Dir an, was die unausgefüllten Rauten in einer Beziehung bedeuten. (Es gibt auch ausgefüllte.)**

**4. Recherchiere die Pfeile.**

**5. Was soll das # Zeichen vor dem setID() ?**

**6. Was hat es mit der Kursivschreibung und dem abstract auf sich?**

Informieren Sie sich mit Hilfe der Unterlagen von Herrn Weiß über die objektorientierte Programmierung und halten Sie Fragen schriftlich fest. Nutze dafür die **beigelegten Kontrollfragen.**

1. Wiederhole indem Du das bereits erstellte Klassendiagramm Mitarbeiter ansiehst und auch den Quellcode dazu. Stelle sicher, dass Du den Begriff Datenkapselung verstehst: Überpüfe, ob deine setID Methode 4-stellige positive IDs sicherstellt. (Ignoriere erstmal die kursiv geschriebene einkommen-Methode – dazu kommen wir später.)
2. Erarbeite das Thema Konstruktoren (das sind die Methoden, die so heißen wie die Klassen) und ersetze den Default Konstruktor in der Klasse Mitarbeiter durch einen, der die Eigenschaften belegt.
3. Erarbeite die Bedeutung der Beziehung zwischen Abteilung und Mitarbeiter. Was -bedeutet das \*? Achtung es taucht 2x auf! Lasse dabei erstmal die Gehaltsliste und den leiter weg – die kommen später.
4. Erstelle die Klasse Abteilung mit der Beziehung zur Klasse Mitarbeiter.
5. Jetzt recherchiere diese merkwürdigen Pfeile, die in die falsche Richtung zu zeigen scheinen. Implementiere anschließend die Klassen SchichtArbeiter, BueroArbeiter und Manager. Du wirst beim implentieren feststellen, dass Du etwas neues zu Konstruktoren lernst. 😉 Und nur Mut – füge die unterschiedlichsten Mitarbeiter Deiner Abteilung zu. Jetzt kannst Du auch einen Leiter hinzufügen und die dazugehörigen Methoden. Wenn eine Leiter ausgetauscht wird, wird der alte Leiter zurückgegeben.
6. Das # vor dem setID hat eine Bedeutung. Recherchiere sowohl das Zeichen wie auch das Thema packages. Lege alle Mitarbeiterklassen in ein package mit Namen mitarbeiter. Die Main-Klasse sollte im default-package bleiben. Implementiere jetzt die Raute und teste die Wirksamkeit in der Main-Klasse.
7. Fast geschafft! 😊 Recherchiere die Bedeutung der Kursivschreibung und des Schlüsselwortes abstract in der Mitarbeiter-Klasse. Füge die Methode einkommen() der Mitarbeiterklasse hinzu und erstelle die Gehaltsliste in der Klasse Abteilung.

**Aufgabe 2**

Bevor wir mit der Speicherung der Mitarbeiter anfangen – die Leute von Aix-Car wollen auch unterschiedliche Kfz in ihrer Software berücksichtigen.

1. Erstelle zuerst mal das **Klassendiagramm**: Eine Klasse Kfz hast Du ja schon. Du musst Sie nur etwas anpassen.

Gesprächsprotokoll des Kundengesprächs

Anwesend: Herr Weiß und Frau Kallenbring

**Fahrer** sind **Schichtarbeiter**, die einen Führerschein haben. Festgehalten wird die Führerscheinklasse. Man kann einen Fahrer anweisen zu einer GPS-Position zu fahren. Dafür muss er wissen, welches Fahrzeug er benutzen soll und welche Position er ansteuern soll.

Ein **Kfz** hat einen maximalen und einen tatsächlichen Tankinhalt. Ein Kfz kann mit Angabe eines Benzinvolumens aufgetankt werden – aber nie mehr als die Tankgröße. Wird zu viel getankt, läuft der Tank über. Das überlaufende Benzinvolumen wird zurückgegeben. Zudem hat es eine aktuelle GPS-Position. Man kann es zu einer neuen GPS-Position fahren. Es kann nur gefahren werden, wenn der Motor an ist und sich ein Fahrer im Kfz befindet. Man kann einen Fahrer einsteigen lassen. Falls er keinen Führerschein hat schlägt das einsteigen fehl. Beim Parken wird der Motor ausgestellt und der Fahrer steigt aus. Die Auslastung eines Fahrzeugs wird in Prozent angegeben. (1,0 entspricht 100%). Wie die Auslastung ermittelt wird ist von der jeweiligen Art des Kfz abhängig.

Eine **GPS**-Position hat einen Längengrad und einen Breitengrad. Der Längengrad nach rechts von Greenwich laufend in Grad eines Kreises zwischen 0° und <360° angegeben. Der Breitengrad liegt zwischen 90° (Nordpol) und -90° (Südpol).

Ein **Personentransportfahrzeug** ist ein Kfz, hat eine Anzahl Sitze (maximale Passagierzahl) und Passagiere, die immer **Mitarbeiter** sind. Beim **Einsteigen der Passagiere** wird überwacht, ob der **Mitarbeiter** noch reinpasst ohne, dass das Fahrzeug überfüllt ist. Passt er nicht rein schlägt das Einsteigen fehl. Beim **Einsteigen des Fahrers** wird überwacht, ob der Fahrer einen Führerschein besitzt, der ihm erlaubt, Personen zu befördern. Personen können auch aussteigen, dabei wird lediglich die Sitzplatznummer angegeben. Beim Parken steigen zudem, was bei allen KFZ passiert auch alle Passagiere aus. Die Auslastung eines Personentransportfahrzeugs ist der Anteil der belegten Sitze.

Ein **Bus** ist ein Personentransportfahrzeug. Der Fahrer muss einen Führerschein haben, der ihn berechtigt einen Bus zu fahren.

**Lkws** haben eine Ladefläche mit einer bestimmten Größe in Quadratmetern. Lkws können beladen und entladen werden, wobei immer berücksichtigt wird, dass die maximale Ladefläche nicht überschritten werden darf und dass nicht mehr entladen werden kann als auf der Ladefläche steht. Außerdem soll die aktuelle Auslastung der Ladefläche als Prozentwert abrufbar sein.

Die **Unternehmensverwaltung** hat eine Liste aller Fahrzeuge (Fuhrpark), aller Mitarbeiter (Personal) und aller Abteilungen. Zusätzlich werden die Schichtarbeiter nicht nur in der Persolnalliste sondern auch in einer zusätzlichen Liste geführt, damit, damit die anpassung der Stundenzahl einfacher verwaltet werden kann.

1. Recherchiere das Thema Überladung (overloading) und das Thema Komposition (vs. Aggregation). Passe anschließend nochmal die Namen Deiner Methoden an und eine der Beziehungsrauten an.

**...**

**7. Überladung**

**8. Komposition**

1. Und jetzt implementiere Dein Modell in Java.

**Aufgabe 3**

Die Toolbox ist jetzt endlich so voll, dass wir uns mit dem Speichern in Dateien befassen können.