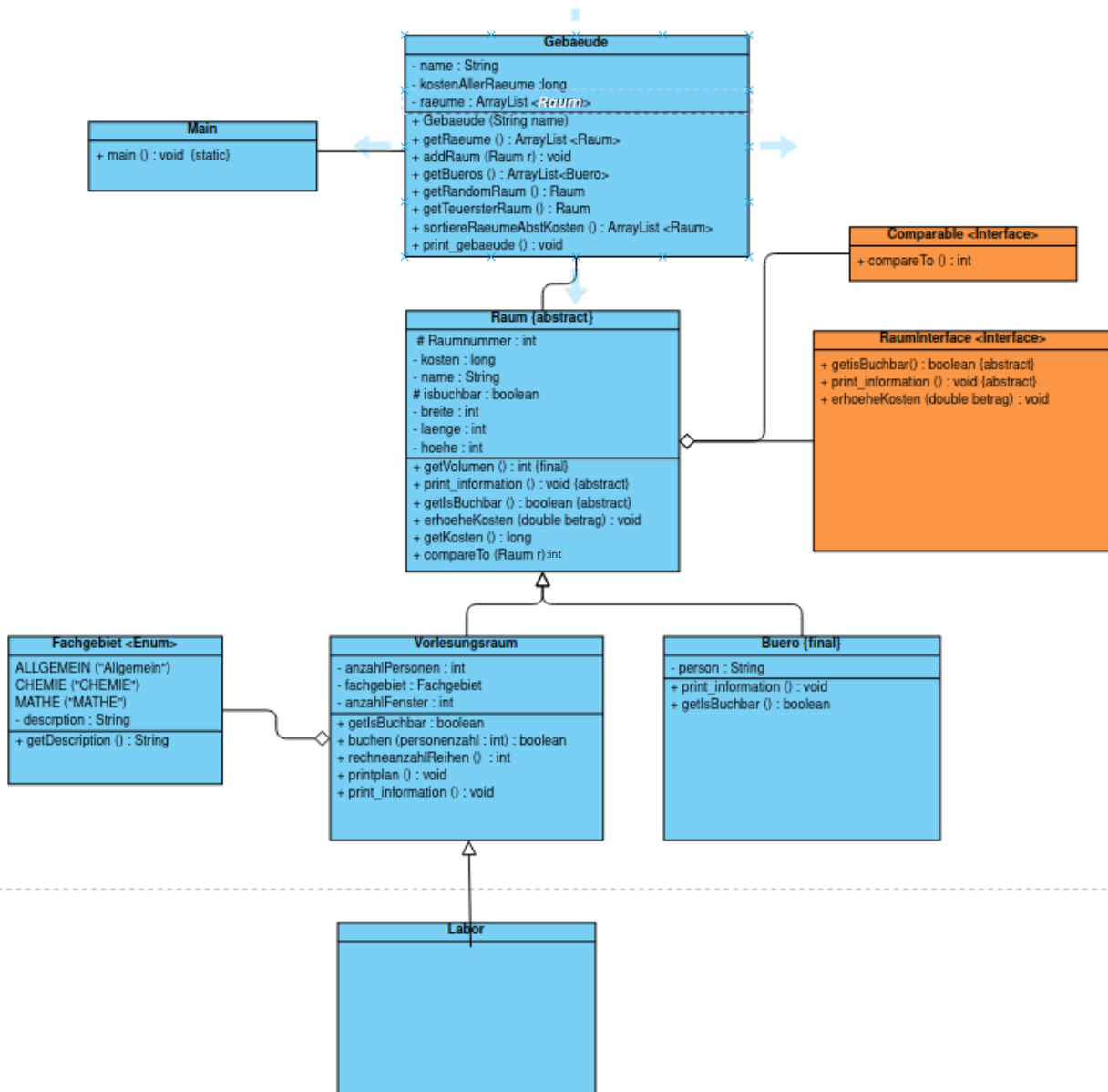


Probeklausur:

Gebäude Klassendiagramm:



Setze das Klassendiagramm komplett um:

Klasse Gebäude :

- Der Konstruktor soll den Name des Gebäudes herübergeben und die kosten auf 0 setzen
- Die Methode `addRaum (Raum r)` soll dem Gebäude einen Raum hinzufügen und die Kosten erhöhen

- Die Methode getBueros soll alle Bueros zurückgeben (instanceof wird verwendet – Steigers Kurs kann es weglassen)
- Die Methode getRandomRaum gibt einen zufälligen Raum aus.
- Die Methode sortiereRaeumeAbstKosten sortiert die Räume nach den Kosten (mittels Comparable)
- Die Methode getTeuersterRaum gibt den teuersten Raum mithilfe der Methode sortiereRaeumeAbstKosten aus -> Exception soll geworfen werden wenn kein Raum gefunden wird
- Die Methode print_gebauede soll mit einer while schleife alle Räume durchlaufen und die Methode print_information auf der Konsole ausgeben

Klasse Raum:

- Der Konstruktor soll alle Attribute initialisieren
- Die Methode CompareTo ist nur für Steffens Kurs relevant und soll den Räume absteigend nach de Kosten sortieren

Klasse Vorlesungsraum:

- Der Konstruktor soll alle Attribute außer die Anzahl Fenster als Parameter übergeben bekommen. Die Anzahl der Fenster ist ein Zufallswert zwischen 0 und 6 und soll im Konstruktor initialisiert werden.
- Die Methode RechneanzahlReihen soll die Anzahl der vollen Reihen errechnen. Die Anzahl der Reihen soll immer die abgerundete Wurzel der Personenzahl sein.
- Die Methode printplan () soll dann den Sitzplan an der Konsole ausgeben. Nutze dafür die oben erstelle Methode, um die Anzahl der vollen Reihen zu errechnen:

Bsp fuer 10 Personen:

```
#      #      #
#      #      #
#      #      #
#      Anzahl Fenster: 1
```

- Die Methode Print_information() soll den Plan sowie die Anzahl der Fenster als auch Name und Nummer ausgeben
- Die Methode GetIsBuchbar ist bei Vorlesungsräumen immer True
- Die Methode Buchen (int anzahlPersonen) soll schauen, ob der Raum genügend Kapazität für die Buchung hat. Wenn nicht, dann soll false zurückgegeben werden.

Klasse Labor:

Die Klasse Labor wird immer das Fachgebiet Chemie zugewiesen und der Konstruktor initialisiert alle Attribute.

Klasse Buero:

- Bueros dürfen nicht gebucht werden und so ist die Methode `getIsBuchbar` immer `false`;

Der Konstruktor initialisiert alle Attribute

Für Steffens Kurs:

Implementiere ebenfalls das Interface `Rauminterface` und binde das `Comparable` Interface in der Raumklasse so ein, dass Räume absteigend nach den Kosten sortiert werden.

Bei der Methode `getTeuerster Raum` kann eine Exception geworfen werden, wenn kein Element in der Raumliste drin ist. Fange die geworfene Exception auf und gebe in dieser Methode auf der Konsole „Es gibt keinen Raum im Gebäude“ aus, falls dies der Fall ist.