



Lernstoff:

- **Verwendung von ArrayList/LinkedList und Hashmap**
- **equals sinnvoll anwenden**
- **Parse von Strings**
- **Runtime Exceptions anwenden (werfen, abfangen, weitergeben)**

Schreibe eine Java Anwendung, in der du die Klassen **Subject**, **Pupil** und **School**, die unten beschrieben sind, implementierst und Parameterprüfungen mit Hilfe von **RuntimeExceptions** löst

Die Klasse **Subject** hat die Attribute (behandle inkorrekte Parameterwerte mit Hilfe einer **IllegalArgumentException** mit aussagekräftiger Message)

- *mark* – die Note 0 bedeutet dabei „nicht beurteilt“
- *name* – darf nicht null oder leer sein

weiteres

- einen Konstruktor, der Werte für die beiden Attribute bekommt und
- die *toString* Methode

Die Klasse **Pupil** hat die Attribute (behandle falsche Benutzereingaben mit Hilfe einer **IllegalArgumentException** mit aussagekräftiger Message)

- *name* – dieser besteht aus Vornamen und Nachnamen – muss daher aus mindestens 2 Namen bestehen – z.B: “Max Muster“
- *dateOfBirth* – Datumswert (Der Schüler muss mindestens 10 Jahre alt sein)
- *schoolClass* – darf nicht null oder leer sein und muss mit einer Zahl beginnen
- *subjects* – eine Collection (ArrayList oder LinkedList) mit sämtlichen Fächern des Schülers (muss mindestens 3 Fächer enthalten).

Es gibt nur einen Konstruktor, der alle Attribute initialisiert und im Fehlerfall (=falscher Attributwert) die RuntimeException ignoriert, damit sie vom aufrufenden Programm behandelt wird, und folgende Methoden:

- *average()* gibt die Durchschnittsnote des Schülers zurück (kann auch keine ganze Zahl sein!) – ist ein „nicht beurteilt dabei“, so wird -1 zurückgegeben.
- *bestSubjects()* gibt eine Collection mit den Fächern zurück, in denen der Schüler am besten ist. Bedenke dabei, dass die beste Note eines Schülers nicht unbedingt 1 sein muss! – d.h. du musst zuerst herausfinden, welche seine beste Note ist (Minimumssuche!), damit du dann diese Fächer in einer *ArrayList* zurückgeben kannst. Achtung! Die beste Note kann nicht 0 sein, denn das wäre ja „nicht beurteilt“!
- *delSubject(Subject subject)* – löscht das angegebene Fach aus der *subjects* Collection falls mehr als 3 Fächer in der Collection enthalten sind (überschreibe die *equals* Methode entsprechend) und gibt bei Erfolg das gelöschte Fach, anderenfalls null zurück.
- *toString()* gibt eine Beschreibung des Schülers inkl. seines Alters und seiner Fächer zurück.
- *getKey()* – gibt die eindeutige Bezeichnung des Schülers zurück, die sich aus *name+schoolClass* zusammensetzt

Die Klasse **School** hat als Attribut:

- eine *HashMap* namens *pupils*, in der die Schüler der Klasse unter ihrem eindeutigen Schlüssel (*name+schoolClass*) gespeichert werden.

... und die folgenden Methoden:

- *addPupil (String name, String schoolClass, LocalDate dateOfBirth, List<Subject>)* :
 - generiert den Schüler und fängt mögliche Exceptions ab. Nur von einem gültigen Schüler wird der entsprechende Schlüssel erstellt (Methode *getKey()*). Ein gültiger Schüler wird in die *HashMap* eingetragen und der Schlüssel des Eintrags von der Methode zurückgegeben falls nicht bereits ein Schüler mit diesem Schlüssel existiert. Bei Misserfolg gibt die Methode *null* zurück.
- *delPupil(String key)*- entfernt den Schüler mit dem angegebenen Schlüssel und gibt den Schüler (oder *null*) zurück.
- *getPupil(String key)* – gibt den Schüler mit dem gegebenen Schlüssel zurück oder *null* falls dieser nicht existiert
- *display()* – zeigt alle Schüler der Schule an

Die Klasse **Test** hat

... die Methoden:

- *generatePupil(String s)* generiert mit Hilfe des übergebenen Strings einen Schüler. Die Daten des Schülers sind wie folgt im String angeordnet – zuerst kommt der Name, der aus 2 Strings besteht, dann folgt die Klasse, dann das Schlüsselwort „Faecher“ und anschließend durch Spaces getrennt die Fächer in der Form *Fach:Note*. Z.B.:

Hans Huber 2EHIF Faecher Mathematik:1 Deutsch:2 POS:4 Englisch:2
Biologie:4 Geographie:0 BSP:1

Oder

Anna Schmied 2AHIF Faecher Mathematik:3 Deutsch:1 POS:1 Englisch:2
Biologie:4 Geographie:3 BSP:1

Das Geburtsdatum wird zufällige gewählt sodass der Schüler 10 bis 15 Jahre alt ist (=> das heutige Datum minus 10 Jahre und minus 1 bis 365*5 Tage). Alle möglichen Exceptions werden abgefangen und die Meldung auf dem Bildschirm ausgegeben.

- *main()* wo du
 - eine Schule *schule* erzeugt.
 - *generatePupil (String s)* mit 2 unterschiedlichen Fehlern aufruft und überprüft ob die ausgegebene Fehlermeldung für dich als Benutzer Sinn macht
 - 3 gültige Schüler mit Hilfe der Methode *generiereSchueler()* generierst und diese in sie *schule* einfügst. Merke dir dabei den Schlüssel des ersten Schülers und gib ihn aus
 - die Schüler der *schule* mit Hilfe der Methode *display()* anzeigen.
 - den ersten Schüler 2 Mal löscht (...sollte nur einmal möglich sein)
 - die Schüler wieder mit Hilfe der *display ()* Methode anzeigen – der erste Schüler (und nur dieser!!!) sollte nun fehlen
 - vom ersten Schüler die Durchschnittsnote und seine besten Noten anzeigen, um die Methoden *bestSubjects()* und *average()* zu testen.
 - zum Schluss noch die Methode *delSubject(String subject)* testest