

Wiederholungsprojekt:

Erstelle ein neues Projekt `Schulverwaltungssoftware`.

1. 📖 Bearbeite Schulbuch S.10, 1.a (Klassendiagramm)
2. 📖 Bearbeite Schulbuch S.10, 1.b (Spezialisierung)
3. 💻 Erstelle ein Projekt `Schulsoftware`.
 - a) Erstelle die Klasse `RAUM` gemäß dem Klassendiagramm mit Konstruktor, Getter- und Setter-Methoden für beide Attribute und einer Methode `infosAusgeben` für eine ordentliche Bildschirmausgabe.
 - b) Erstelle die Klasse `KLASSENZIMMER` als Unterklasse von `RAUM` mit dem zusätzlichen Attribut `klasse`. Implementiere den Konstruktor mit Hilfe von `super`, außerdem die fehlende Getter- und Setter-Methode sowie eine umgeschriebene und ergänzte Methode `infosAusgeben`, die ohne `super` abläuft.
 - c) Erstelle die Klasse `FACHRAUM` als Unterklasse von `RAUM` mit dem zusätzlichen Attribut `fach`, wieder mit Konstruktor, Getter- und Setter-Methode und wieder einer umgeschriebenen Methode `infosAusgeben`, die mit `super` arbeitet.
 - d) Erstelle die Klasse `SCHULVERWALTUNG`, die eine `ArrayList` `räume` vereinbart und initialisiert und die Methoden `raumHinzufügen` sowie `infosAusgeben` zur Bildschirmausgabe aller Infos zu den Räumen hat.
 - e) Erkläre kann den Begriff Polymorphismus und das „Überschreiben einer Methode“ am Beispiel aus d).
 - f) Erstelle die `TESTKLASSE` in der du je ein Objekt der Klassen `KLASSENZIMMER`, `FACHRAUM` und `SCHULVERWALTUNG` deklarierst und initialisierst. In einer `testmethode` werden die Räume zur `ArrayList` dazugefügt und veranlasst, dass alle Infos ausgegeben werden.
4. 💻 Speichere das Projekt unter neuem Namen `SchulsoftwareFeld` oder öffne das Lösungsprojekt `12_SchulsoftwareArrayList_Opp`.
 - a) Nimm alle Anpassungen in der Klasse `SCHULVERWALTUNG` vor, um das Ganze mit einem `FELD` zu implementieren. Der Konstruktor wartet auf Übergabe eines Wert für die maximale Feldgröße.
 - b) Passe die Testklasse an, sodass das erzeugte Feld die Größe 10 hat.