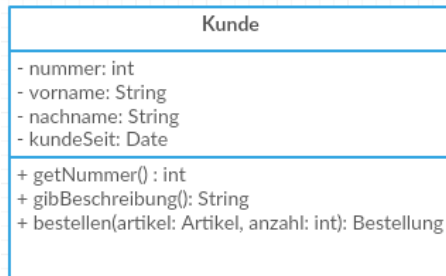


1. Klassen und Objekte

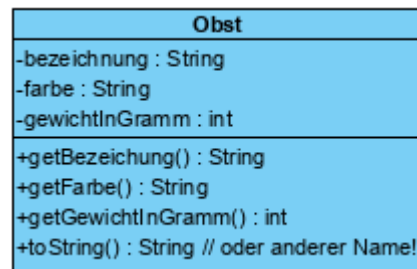
[LZ 1.2a] Erläutern sie folgendes Klassenmodell:



- die Klasse „Kunde“ wird definiert
- „Kunde“ (und damit jedes Kundenobjekt) hat private Attribute:
 - ein Attribut „nummer“ vom Typ „int“
 - ein Attribut „vorname“ vom Typ „String“
 - ein Attribut „nachname“ vom Typ „String“
 - ein Attribut „kundeSeit“ vom Typ „Date“
- sowie folgende öffentlich nutzbaren Methoden:
 - getNummer: parameterlos vom Ergebnistyp „int“
 - gibBeschreibung: parameterlos vom Ergebnistyp „String“
 - bestellen: 2 Parameter vom Typ „Artikel“ bzw. „int“ vom Ergebnistyp „Bestellung“

[LZ 1.2b] Entwerfen Sie ein UML-Klassenmodell zu folgenden Aussagen:

- Obst hat eine Bezeichnung (z.B. „Apfel“), eine Farbe (z.B. „rot“) und ein Gewicht in Gramm (z.B. 100)
- die Eigenschaften (s.o.) können über Methoden der Art get<Eigenschaft> gelesen werden
- Jedes Obst-Objekt kann eine Beschreibung als String liefern



1. Klassen und Objekte

[LZ 1.2c] Stellen Sie folgende Obst-Objekte grafisch dar

- 150gr. schwerer grüner Apfel
- hellgrüne Birne, Gewicht: 120 gr.
- gelbe Banane, 200 gr.
- rote Kirsche, 2 gr.

Apfel: Obst
-bezeichnung : String = "Apfel"
-farbe : String = "grün"
-gewichtInGram : int = 150

Birne: Obst
-bezeichnung : String = "Birne"
-farbe : String = "hellgrün"
-gewichtInGram : int = 120

Banane: Obst
-bezeichnung : String = "Banane"
-farbe : String = "gelb"
-gewichtInGram : int = 200

Kirsche: Obst
-bezeichnung : String = "Kirsche"
-farbe : String = "rot"
-gewichtInGram : int = 2

1. Klassen und Objekte

[LZ 1.3] Java-Klassen definieren und nutzen können



Definieren Sie eine Klasse Person

- eine Person hat ein Alter, z.B. 29
- eine Person hat einen Namen, z.B. „Amelie Müller“
- eine Person wird bei Erzeugung initialisiert über die Parameter für Name und Alter
- eine Person kann sich auf den Bildschirm ausgeben (gleichnamige Methode!), z.B. „Name: Amelie Müller, Alter: 29“

Ergänzen Sie folgendes Hauptprogramm zum Testen

```
static public void main(String[] args) {
    Person peter = new Person("Peter", 20);
    Person amelie = new Person("Amelie", 29);

    peter.ausgeben();
    amelie.ausgeben();
}
```

siehe Kursraum Ordner Übungen/klassenUndObjekte.zip – Klasse Person

1. Klassen und Objekte

[LZ 1.8] Den Java-Parameterübergabemechanismus kennen und einsetzen können

- [*] Geben Sie den Methodenprototypen zu folgendem Aufruf von `erzeugeKampagne` an:

```
String[] details = new String[] { "detail 1", "detail 2"};
Auftrag a1 = new Auftrag("Businessplan-Erstellung");
Firma f1 = new Firma("Startup One");
String ergebnis = erzeugeKampagne(a1, f1, 2017, details);
```

`String erzeugeKampagne(Auftrag a, Firma f, int, String[] details)`

- Sie möchten analog zu „C“ einen `int`-Wert als Ergebnisparameter zurückgeben (nicht über das Methodenergebnis!). Ein Beispiel wäre die „C“-Methode

```
void add(int x, int y, int *summe) {
    *summe = x + y;
}
```

Der Aufruf `add(3, 5, &erg);` bewirkt, dass die Variable `erg` den Wert 8 erhält.
Geben Sie eine **Java-Lösung für obige add-Methode** an.

siehe Kursraum Ordner Übungen/klassenUndObjekte.zip – Klasse Parameteruebergabe

1. Klassen und Objekte

[LZ 1.10] Die Sichtbarkeitsstufen kennen und deren Anwendung beherrschen

- Gegeben sei folgende Klasse – **ergänzen Sie die Sichtbarkeiten!**

```
public class Film
    private String titel;
    private String nameRegisseur;
    private Schauspieler[] mitwirkende;

    static public void main(String[] args) {
        Film starWars7 = new Film();
    }

    public String toString() {
        return "Titel: " + titel +
            ", Regie: " + nameRegisseur;
    }
}
```



1. Klassen und Objekte

[LZ 1.11] Die Array-Syntax kennen und Java-Arrays einsetzen können

- **[*]** Definieren Sie ein char-Feld und belegen Sie es mit den Großbuchstaben von ‚A‘ bis ‚Z‘. Berechnen Sie anschließend die Summe der ASCII-Werte der Buchstaben und geben Sie die Buchstaben und die Summe aus.

siehe Kursraum Ordner Übungen/klassenUndObjekte.zip – Klasse SummeAsciiWerte



1. Klassen und Objekte

[LZ 1.12] Die Java-String API kennen und diese in Java-Programmen einsetzen können

- [*] Definieren Sie ein Feld von Strings und belegen Sie es mit den Werten „Java Strings“, „ist sind“, „cool toll“. Geben Sie anschließend nur die erste Hälfte jedes Strings aus, danach jeweils die 2. Hälfte jedes Strings und abschließend nur die Strings aus, die das Wort „cool“ enthalten.

siehe Kursraum Ordner Übungen/klassenUndObjekte.zip – Klasse StringFeld

1. Klassen und Objekte

[LZ 1.13] Wissen, wie Java-Objekte inhaltlich verglichen werden können

- **[*]** Definieren Sie eine Rechteck-Klasse, die einen Vergleich von Rechteck-Objekten anhand ihrer Fläche erlaubt.
siehe Kursraum Ordner Übungen/klassenUndObjekte.zip – Klasse Rechteck

1. Klassen und Objekte

[LZ 1.14] Motivation und Syntax statischer Elemente kennen und anwenden

- **[*]** Definieren Sie ein statisches Attribut „zaehler“ zu einer Klasse „X“ und speichern Sie darin die Anzahl der Objekte der Klasse X. Über einen Getter (getZaehler()) soll der Zähler auslesbar sein (z.B. für eine Ausgabe)
siehe Kursraum Ordner Übungen/klassenUndObjekte.zip – Klasse X