

Übungen zur Vorlesung „Programmieren“

Thema: Objektorientiertes Programmieren

1. Öffnen Sie in PyCharm den Ordner `oop`. Dieser Ordner enthält einen Ordner mit dem Namen `src`. In diesem Ordner befinden sich drei Dateien, `account.py`, `customer.py` und die `main.py`. In der `account.py` ist die Klassendefinition für `Account` hinterlegt und in `Customer` die für die `Customer`. Die `main.py` stellt den Startpunkt unserer Anwendung dar.

Hinweis:

Der `src`-Ordner muss als `Source-Ordner` angegeben werden. Dies ist nötig, damit der Import der `Customer`-Klasse funktioniert. Dazu müssen Sie auf den `src`-Ordner einen Rechtsklick ausführen und im Menüpunkt „Mark Directory as“ den Unterpunkt „Sources Root“ auswählen. Der Ordner wird anschließend bläulich eingefärbt.

Hinweis:

Es sollte pro Projekt nur ein `Sources Root` geben. Sollten Sie sich für ein neuen `Sources Root` entscheiden, müssen Sie die Markierung des aktuellen `Sources Root` entfernen.

2. Folgende Fragen beziehen sich auf die Codezeilen in der `main.py`.
 - a. Was passiert durch die Angabe `Customer(„Max“, „Mustermann“)`? Welche Auswirkung hat die Angabe von `Max` und `Mustermann`? Betrachten Sie dazu die Klasse `Customer`, die sich in der Datei `customer.py` befindet.
 - b. Was ist das Ergebnis, wenn die Klammer von `Customer` leer bleibt, also: `Customer()`?
 - c. Zeile 8 erzeugt eine neue Instanz vom Typ `Account` und wird dem `Customer` `Max Mustermann` zugewiesen. Welches Attribut in `Account` gewährleistet die Eigentümerschaft?
 - d. In Zeile 10 wird die Anweisung `a1.deposit(100.0)` und in der Zeile 11 wird `a1.display()` ausgeführt. Welche Auswirkungen hat die Methode `deposit` auf das Konto?
 - e. Die Zeilen 11-12 erzeugen zwei Konsolenausgaben. Wo liegen die Unterschiede in der Ausgabe? Was vermuten Sie ist der Grund dafür, dass sich die Ausgaben unterscheiden?
 - f. In den Zeilen 13 und 14 werden die beiden Objekte an die `print()`-Methode übergeben. Was bedeutet die Ausgabe in der Konsole?
3. Alle nachfolgenden Fragen ab Codezeile 17 einfügen.
 - a. Erzeugen Sie einen weiteren `Customer` mit dem Namen `Hildegard Mustermann`.
 - b. Initialisieren Sie einen `Account` `a2` und übergeben das `Customer`-Objekt `Hildegard Mustermann`, sodass Frau `Mustermann` die Eigentümerin von `a2` ist.
 - c. Experimentieren Sie weiter, führen Sie Buchungsvorgänge auf den Konten durch und lassen Sie sich die Ergebnisse auf der Konsole ausgeben.
4. Wenn Sie die beiden Klassen `Account` und `Customer` betrachten, welche weitere Funktionalitäten wären wünschenswert? Verbalisieren Sie ihre Antwort, es ist kein Programmcode nötig.