

# Lernfeld 11a – Funktionalität in Anwendungen realisieren

## Lernsituation 11a.1 : Vertiefung von OOP-Konzepten



### Handwerker & Anlagenservice – die HASE GmbH

Vertiefung der OOP-Konzepte

✕

---

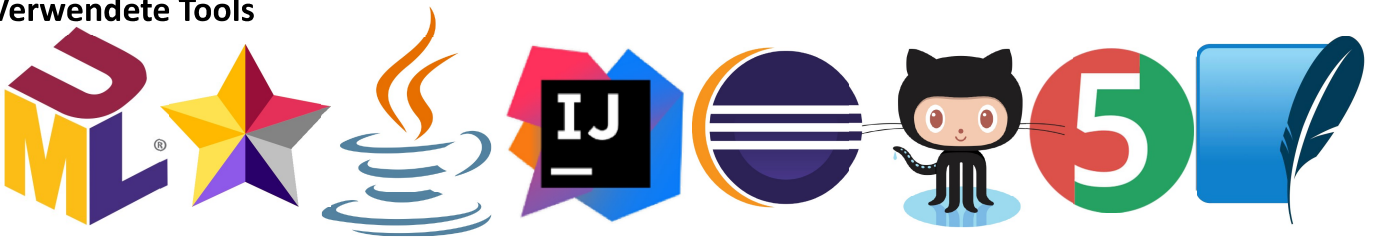
Schritt 1 von 6

<p><b>1. Grundlagen der Programmierung</b></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">Struktogramme, Programmablaufpläne, Compiler und Interpreter, Algorithmen</div>	<p><b>4. Von Kunden und Firmen</b></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">Vererbung, abstrakte Klassen, Schnittstellen modellieren und implementieren</div>
<p><b>2. Kundenverwaltung kennenlernen</b></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">Repository, Datenstrukturen: ArrayList, HashMap</div>	<p><b>5. Persistente Datenspeicherung</b></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">Exception Handling und Datenbanken</div>
<p><b>3. Aufträge &amp; Adressen hinzufügen</b></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">Assoziationen modellieren und implementieren</div>	<p><b>6. Generische Programmierung</b></p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">Generische Klassen und Methoden</div>

### Rahmenhandlung

Der Prozess der Kundenverwaltung wird digitalisiert und ein erster Prototyp einer GUI (Graphical User Interface) wurde erstellt (siehe LF10a). In LF11a wird das passende Backend geplant und als Java-Konsolenprogramm programmiert, inkl. Datenhaltung.

### Verwendete Tools



**Die Lernenden verfügen über die Kompetenz, modulare Komponenten zur Informationstechnischen Verarbeitung von Arbeitsabläufen und Geschäftsprozessen zu entwickeln und deren Qualität zu sichern.**

- Die Azubis leiten aus den Informationsobjekten der vorgegebenen Prozess-beschreibungen die dazu notwendigen Datenstrukturen, z.B. ArrayList, HashMap und Funktionalitäten ab, z.B. Klassen mit Vererbung, generics, Assoziationen
- Sie planen modulare Softwarekomponenten und beschreiben deren Funktionsweise mit Diagrammen und Modellen, z.B. Klassendiagramm.
- Die Schülerinnen und Schüler realisieren, auch im Team, z.B. mit git, die Softwarekomponenten und binden diese an Datenquellen an, z.B. mit SQL. Sie dokumentieren die Schnittstellen.
- Sie testen der erforderliche Funktionalität, indem sie Testfälle formulieren und automatisierte Testverfahren anwenden, z.B. Unit-Tests
- Die Schülerinnen und Schüler beurteilen die Funktionalität anhand festgelegter Kriterien und leiten Maßnahmen zur Überarbeitung der erstellten Module ein.