**Voraussetzungen / Lernziele:**

**+ Rechnerbenutzung**

*# Mails (Übung 1, WS03, Aufgabe 1)*

*# Recherchieren (Übung 1, WS03, Aufgabe 2 + 3)*

*# Deckblatt erstellen (Übung 1, WS03, Aufgabe 4)*

*# GDI-Anmeldung (Übung 1, WS03, Aufgabe 5)*

**+ Umgang mit IDEs / Programm zum Laufen bringen, Fehler beheben**

*# Programm zum Laufen bekommen (Übung 2, WS10-WS12, Aufgabe 2)*

* *Übung 2, WS06-WS08, Aufgabe 3*
* *Übung 3, WS03-WS04, Aufgabe 3*

**+ Elementare Datentypen, Operatoren, Ausdrücke / Ausdrücke und Operatoren (besser) verstehen**

*# Ausdrücke auswerten (Übung 2, WS02-WS04, WS06-WS08, WS10-WS12, Aufgabe 1)*

* *Übung 3, WS12, Aufgabe 1*

*# Ausdrücke programmieren (Übung 3, WS03-WS04, Aufgabe 1)*

* *Baut auf „Ausdrücke auswerten“ (Übung 2, WS02-WS04, WS06-WS08, WS10-WS12, Aufgabe 1) auf*

*# Ausdrücke erklären (Übung 3, WS03-WS04, Aufgabe 2)*

**+ Elementare Datentypen, Operatoren, Ausdrücke, Konsolenausgabe / Ausdrücke und Operatoren (besser) verstehen**

*# Ausdrücke programmieren (Übung 2, WS02, Aufgabe 4 + 5, WS03, Aufgabe 3)*

* *Baut auf „Ausdrücke auswerten“ (Übung 2, WS02, Aufgabe 1) auf*
* *Aufgabe 5 baut (zusätzlich) auf „Ausdrücke programmieren“ (Übung 2, WS02, Aufgabe4) auf*

*# Osterprogramm (Übung 2, WS09, Aufgabe 2)*

* *Übung 3, WS08, WS12, Aufgabe 2*
* *Übung 3, WS11, Aufgabe 3*
* *Übung 4, WS10, Aufgabe 4*

*# Zahlen-Palindrom (Übung 2, WS09, Aufgabe 3)*

* *Übung 3, WS08, Aufgabe 3*

*# Palindrom (Übung 3, WS10-WS11, Aufgabe 2)*

* *Übung 4, WS12, Aufgabe 2*

*# Ausdrücke verstehen (Übung 2, WS02-WS04, Aufgabe 2 + 3)*

* *Übung 2, WS06-WS07, Aufgabe 2*

*# Wochentagsberechnung (Übung 3, WS10, Aufgabe 3)*

**+ Konsolen Ein- und Ausgabe, Bedingungen, Schleifen, if-Anweisung**

*# Zeichen einlesen (Übung 3, WS02, Aufgabe 1)*

* *Übung 3, WS09, Aufgabe 3*
* *Übung 6, WS03, Aufgabe 1*
* *Übung 6, WS07, Aufgabe 3*
* *Übung 7, WS06, Aufgabe 2*

*# Programm umschreiben (Übung 3, WS02, Aufgabe 2)*

* *Übung 5, WS03, Aufgabe 1*
* *Übung 8, WS04, Aufgabe 1*

*# Summierer (Übung 3, WS02, Aufgabe 3)*

* *Übung 2, WS09, Aufgabe 1*
* *Übung 3, WS06-WS08, Aufgabe 1*
* *Übung 4, WS04, Aufgabe 1*
* *Übung 5, WS03, Aufgabe 2*

*# Summierer (erweitert) (Übung 3, WS02, Aufgabe 4)*

*# Zahlen zerlegen (Übung 3, WS02, Aufgabe 5)*

* *Übung 5, WS09, Aufgabe 2*
* *Übung 6, WS03, Aufgabe 2*

*# Divisor (Übung 3, WS12, Aufgabe 3)*

* *Übung 3, WS10-WS11, Aufgabe 1*

**+ Entwurf/Verständnis von Algorithmen, Bedingungen, Schleifen, if-Anweisung, Konsolen Aus- und Eingabe**

*# Primzahlen (Übung 3, WS09, Aufgabe 1)*

* *Übung 4, WS06-WS08, Aufgabe 1*
* *Übung 5, WS04, Aufgabe 1*

*# Sieb des Eratosthenes (Primzahlen) (Übung 3, WS02, Aufgabe 6)*

* *Übung 4, WS06-WS08, Aufgabe 2*
* *Übung 5, WS04, Aufgabe 2*

*# Fibonacci-Folge (Übung 4, WS10, Aufgabe 1)*

* *Übung 6, WS11, Aufgabe 1*

**+ Schleifen, Eindimensionale Arrays**

*# Sieb des Eratosthenes (Primzahlen) ändern (Übung 4, WS02, Aufgabe 1)*

* *Baut auf „Sieb des Eratosthenes (Primzahlen)“ (Übung 3, WS02, Aufgabe 6) auf*
* *Übung 3, WS09, Aufgabe 2*

*# Sieb des Eratosthenes (Primzahlen) mit Array*

*(Übung 4, WS02, Aufgabe 2)*

* *Optional: Baut auf „Sieb des Eratosthenes (Primzahlen)“ (Übung 3, WS02, Aufgabe 6) auf*

*# args-Parameter abfragen (Übung 4, WS02, Aufgabe 3)*

* *Übung 5, WS08, Aufgabe 1*
* *Übung 7, WS11, Zusatzaufgabe 3*

*# Character-Array (char to int) (Übung 4, WS02, Aufgabe 4)*

*# Schachbrett-Legende (Übung 4, WS10, Aufgabe 2)*

*# Kleines Adressregister (Übung 4, WS10, Aufgabe 3)*

* *Übung 6, WS11, Aufgabe 2*

**+ Dateihandling**

*# Hello World Programm (Übung 8, WS11, Aufgabe 3)*

**+ Mehrdimensionale Arrays, Dateihandling**

*# Adress-Array (Übung 5, WS02, Aufgabe 1)*

* *(Baut auf „Adress-Array“ aus Vorlesung auf)*
* *Übung 6, WS03, Aufgabe 3*
* *Übung 6, WS04, Aufgabe 1*

*# Adress-Array (Datenformat ändern) (Übung 6, WS04, Aufgabe 2)*

* *Baut auf „Adress-Array“ (Übung 5, WS02, Aufgabe 1) auf*

*# Pascal’sches Dreieck (Übung 5, WS02, Zusatzaufgabe 2)*

*# DVD Verwaltung (Übung 5, WS06, Aufgabe 1)*

* *Übung 6, WS06, Aufgabe 1*

**+ Kommentare / Kommentare sinnvoll verwenden**

*# Programm kommentieren (Übung 5, WS02, Aufgabe 3)*

* *Baut auf „Adress-Array“ (Übung 5, WS02, Aufgabe 1) auf*

*# Programm kommentieren (Übung 5, WS02, Zusatzaufgabe 4)*

* *Baut auf „Pascal’sches Dreieck“ (Übung 5, WS02, Zusatzaufgabe 2) auf*

*# TicTacToe kommentieren (Übung 7, WS02, Aufgabe 2)*

* *Baut auf Übung 1, WS02 auf (nicht vorhanden)*
* *Übung 8, WS03, Aufgabe 2*

*# Adress-Array kommentieren (Übung 6, WS03, Aufgabe 4)*

* *Baut auf „Adress-Array“ (Übung 6, WS03, Aufgabe 3) auf*

**+ Methoden**

*# Character-Array (char to int) überarbeiten (Übung 6, WS02, Aufgabe 1)*

* *Baut auf „Character-Array (char to int)“ (Übung 4, WS02, Aufgabe 4) auf*

*# Adress-Array überarbeiten (Übung 6, WS02, Aufgabe 2)*

* *Baut auf „Adress-Array“ (Übung 5, WS02, Aufgabe 1) auf*
* *Übung 7, WS03, Aufgabe 2*
* *Übung 8, WS04, Aufgabe 2*

*# S-Bahn-Netz von Byteburg (Übung 4, WS09, Aufgabe 1)*

* *Übung 4, WS11, Aufgabe 2*
* *Übung 6, WS08, Aufgabe 1*
* *Übung 9, WS10, Zusatzaufgabe 1*

*# ISBN-13 Prüfziffer-Berechnung (Übung 4, WS11, Aufgabe 1)*

* *Übung 5, WS10, Aufgabe 2*

*# DVD Verwaltung (Programmteile auslagern)*

*(Übung 5, WS06, Aufgabe 2)*

* *Baut auf „DVD Verwaltung“ (Übung 5, WS06, Aufgabe 1) auf*

*# Flaggen-Koordinaten-System (Übung 5, WS08, Aufgabe 2)*

*# Bacon-Chiffre (Übung 5, WS11, Aufgabe 1)*

* *Übung 8, WS12, Aufgabe 1*
* *Übung 9, WS11, Aufgabe 1*

*# Bacon-Chiffre modifizieren (Übung 9, WS11, Aufgabe 1)*

* *Baut auf „Bacon-Chiffre“ (Übung 5, WS11, Aufgabe 1) auf*

*# Primzahlen Programme zusammenfassen (Übung 6, WS06, Aufgabe 2)*

* *Baut auf „Primzahlen“ (Übung 3, WS09, Aufgabe 1) und „Sieb des Eratosthenes (Primzahlen)“ (Übung 3, WS02, Aufgabe 6) auf*

*# PrimeCalculatorAndPrinter (Übung 8, WS08, Aufgabe 5)*

* *Baut auf „Primzahlen“ (Übung 3, WS09, Aufgabe 1) und „Sieb des Eratosthenes (Primzahlen)“ (Übung 3, WS02, Aufgabe 6) auf*

*# String-to-Integer (Programmteile auslagern)*

*(Übung 7, WS03, Aufgabe 1)*

* *Baut auf „String-to-Integer“ (Übung 3, WS06-WS07, Aufgabe 2) auf*

*# Adress-Arrays zusammenfassen (Übung 7, WS04, Aufgabe 2)*

* *Baut auf „Adress-Array“ (Übung 6, WS04, Aufgabe 1) und „Adress-Array (Datenformat ändern)“ (Übung 6, WS04, Aufgabe 2) auf*

*# Adress-Arrays Zusammenfassung (Programmteile auslagern) (Übung 7, WS04, Aufgabe 3)*

* *Baut auf „Adress-Arrays zusammenfassen“ (Übung 7, WS04, Aufgabe 2) auf*

*# Parkplatzleitsystem (Übung 7, WS11, Aufgabe 1)*

*# q-Funktion (Übung 9, WS09, Aufgabe 3)*

**+ Iteration, Rekursion, Eindimensionale Arrays, Methoden**

*# String als Character-Array implementieren (Übung 7, WS02, WS06, WS08-WS09, Aufgabe 1)*

* *Übung 5, WS07, Aufgabe 1*
* *Übung 8, WS03, Aufgabe 1*
* *Übung 8, WS04, Aufgabe 3*

*# String-Funktionen (Übung 5, WS10, WS12, Aufgabe 1)*

* *Übung 6, WS11, Aufgabe 2*
* *Übung 8, WS11, Aufgabe 1*

**+ Iteration, Rekursion, Eindimensionale Arrays, Methoden, Dateihandling**

*# Adressregister erweitern (Übung 8, WS11, Aufgabe 2)*

* *Baut auf „Kleines Adressregister“ (Übung 4, WS10, Aufgabe 3) auf*

**+ Eindimensionale Arrays, Dateihandling, Klassen / Verständnis von Programm-Strukturen**

*# Adress-Array überarbeiten (Übung 8, WS02, Aufgabe 1)*

* *Baut auf „Adress-Array überarbeiten“ (Übung 6, WS02, Aufgabe 2) auf*

*# Lines of Code Programm (Übung 6, WS12, Aufgabe 2)*

**+ Abstrakte Datentypen, Klassen / Entwurf, Implementierung, Testen von Programmen**

*# ADTs spezifizieren (Übung 9, WS02, Aufgabe 1)*

*# Date und Duration implementieren (Übung 9, WS02, Aufgabe 2)*

* *Baut auf „ADTs spezifizieren“ (Übung 9, WS02, Aufgabe 1) auf*

*# ADT-Programm (Übung 9, WS02, Aufgabe 3)*

**+ Kontrollstrukturen**

*# String-to-Integer (Übung 3, WS06-WS07, Aufgabe 2)*

* *Übung 4, WS04, WS09, Aufgabe 2*
* *Übung 4, WS12, Aufgabe 1*
* *Übung 5, WS03, Aufgabe 3*

*# String-to-Integer (erweitert) (Übung 3, WS06-WS07, Aufgabe 3)*

* *Baut auf „String-to-Integer“ (Übung 3, WS07, Aufgabe 2) auf*
* *Übung 4, WS04, Aufgabe 3*
* *Übung 5, WS03, Aufgabe 4*

*# Quellcode ausgeben (Übung 5, WS09, Aufgabe 1)*

**+ Interfaces, Rekursion / Nutzen und Verwendung von Interfaces verstehen**

*# Strings-Interface implementieren (Übung 6, WS07, Aufgabe 1)*

*# Bibliotheks-Interface erstellen (Übung 6, WS07, Aufgabe 2)*

* *Übung 6, WS10, Aufgabe 1*
* *Übung 7, WS08, Aufgabe 3*
* *Übung 7, WS09, Aufgabe 4*

*# Fahrkarten-Interface (Übung 6, WS10, Aufgabe 2)*

**+ Interfaces**

*# Calculate-Prime Interface (Übung 7, WS07, Aufgabe 1)*

* *Baut auf „Primzahlen“ (Übung 3, WS09, Aufgabe 1) und „Sieb des Eratosthenes (Primzahlen)“ (Übung 3, WS02, Aufgabe 6) auf*
* *Übung 7, WS08-WS09, Aufgabe 2*

*# Calculate-Prime Timer (Übung 7, WS07, Aufgabe 2)*

* *Baut auf „Calculate-Prime Interface“ (Übung 7, WS07, Aufgabe 1) auf*
* *Übung 7, WS09, Zusatzaufgabe 3*
* *Übung 8, WS08, Aufgabe 6*

*# TicTacToe implementieren (Übung 7, WS10, Aufgabe 1)*

*# Hotel-Interface (Übung 7, WS12, Aufgabe 1)*

* *Übung 9, WS11, Aufgabe 3*

*# T-Shirt Versand (Übung 7, WS12, Aufgabe 2)*

* *Übung 9, WS09, Aufgabe 1*

**+ Arrays, Klassen, Konstruktoren**

*# DVD Verwaltung umschreiben (Übung 8, WS06, Aufgabe 1)*

* *Baut auf „DVD Verwaltung“ (Übung 5, WS06, Aufgabe 1) auf*

*# DVD Verwaltung erweitern (Übung 8, WS06, Aufgabe 2)*

*# DVD Verwaltung nochmal erweitern (Übung 9, WS06, Aufgabe 1)*

* *Baut auf „DVD Verwaltung erweitern“ (Übung 8, WS06, Aufgabe 2) auf*

**+ Interfaces, Klassen, Dateihandling**

*# Address Interface implementieren (Übung 8, WS07-WS09, Aufgabe 1)*

*# Date Interface implementieren (Übung 8, WS09-WS10, Aufgabe 2)*

*# Contact Implementierung erweitern (Übung 9, WS07, Aufgabe 1)*

* *Baut auf „Contact Interface implementieren“ (Übung 8, WS07-WS08, Aufgabe 2) auf*

*# Contact Management Implementierung erweitern (Übung 9, WS07, Aufgabe 2)*

* *Baut auf „Contact Management Interface implementieren“ (Übung 8, WS07-WS08, Aufgabe 3) auf*

**+ Interfaces, Klassen**

*# Contact Interface implementieren (Übung 8, WS07-WS08, Aufgabe 2)*

* *Übung 8, WS09, Aufgabe 3*

*# Contact Management Interface implementieren (Übung 8, WS07-WS08, Aufgabe 3)*

* *Übung 8, WS09, Aufgabe 4*

*# Contact GUI (Übung 8, WS07-WS08, Aufgabe 4)*

* *Baut auf „Address Interface implementieren“, „Contact Interface implementieren“ und „Contact Management Interface implementieren“ (Übung 8, WS07, Aufgabe 1, 2 und 3) auf*
* *Übung 8, WS09, Aufgabe 5*

*# Container (Übung 8, WS10, Aufgabe 1)*

*# Suchmethoden Interface implementieren (Übung 9, WS07, Aufgabe 3)*

*# Contact GUI erweitern (Übung 9, WS07, Aufgabe 4)*

* *Baut auf „Contact GUI“ (Übung 8, WS07-WS08, Aufgabe 4) auf*

*# ContactComparator Interface implementieren (Übung 9, WS08, Aufgabe 1)*

* *Baut auf „Contact Interface implementieren“ (Übung 8, WS07-WS08, Aufgabe 2) auf*

*# Contact Manager Implementierung erweitern (Übung 9, WS08, Aufgabe 2)*

* *Baut auf „Contact Management Interface implementieren“ (Übung 8, WS07-WS08, Aufgabe 3) auf*

*# T-Shirt Versand Interface implementieren (Übung 9, WS09, Aufgabe 2)*

**+ Klassen, Objekte**

*# Adressverwaltungs-Programm splitten (Übung 9, WS03-WS04, Aufgabe 1)*

* *Baut auf „Adress-Array überarbeiten“ (Übung 6, WS02, Aufgabe 2) auf*

*# Gesplittetes Adressverwaltungs-Programm erweitern (Übung 9, WS04, Aufgabe 2)*

* *Baut auf „Adressverwaltungs-Programm splitten“ (Übung 9, WS03-WS04, Aufgabe 1) auf*

**+ Switch-Case**

*# Contact GUI Implementierung ändern (Übung 9, WS08, Aufgabe 3)*

* *Baut auf „Contact GUI“ (Übung 8, WS07-WS08, Aufgabe 4) auf*

**+ Mehrdimensionale Arrays**

*# Primzahlen Programme zusammenfassen (erweitern) (Übung 6, WS06, Aufgabe 3)*

* *Baut auf „Primzahlen Programme zusammenfassen“ (Übung 6, WS06, Aufgabe 2)*

*# Acht-Damenproblem (Übung 6, WS09, Aufgabe 1)*

*# Acht-Damenproblem (Lösung ermitteln) (Übung 6, WS09, Zusatzaufgabe 2)*

* *Baut auf „Acht-Damenproblem“ (Übung 6, WS09, Aufgabe 1) auf*

*# Acht-Damenproblem (Anzahl Lösungen ermitteln) (Übung 6, WS09, Zusatzaufgabe 3)*

* *Baut auf „Acht-Damenproblem (Lösung ermitteln)“ (Übung 6, WS09, Zusatzaufgabe 2) auf*

*# Labyrinth (Übung 9, WS12, Aufgabe 1)*

**+ Debugging**

**+ Switch-case**

**+ Rekursion**

**+ Datei Ein- und Ausgabe**

**+ Arrays von Arrays**

**+ Klassen als Datenstruktur**

**+ Abstrakte Datentypen**

**+ Dynamische Datenstrukturen (Listen)**

**+ Objekte, Konstruktoren**

**+ Enum**