Informatik 11

Basisinformationen zu Informatik 11.

Inf11

Grundbegriffe
OOM python
Datenstruktur Liste
Rekursive Datenstruktur Warteschlange und Liste
Datenstruktur Schlange - Trennung von Struktur und Inhalt durch Einführung einer Schnittstelle (Interface)
Unterschied zwischen der Datenstruktur Stapel (engl. Stack) und Schlange (engl. Queue)
Unterschied Rekursion und Iteration
Datenstruktur Baum als spezieller Graph
Spezialfall geordneter Binärbaum
Datenstruktur Graph als Verallgemeinerung der Datenstruktur Baum
Darstellung eines Graphen mit einer Adjazenzmatrix
Softwareentwicklung/ Projektmanagement
Medieninformatik

WET crossmedia | crossmedia-wettbewerb.de
> Sammlung Skizze Entwurf Photo Materialcollage Unikat - Publikation print & web als Plenumsarbeit

> crossmedia | Anmeldeformulare pdf vorausfüllen, print, Rest s ausfüllen > Einreichung. Ein Preisgeld-Gewinn verbleibt in der Medienwerkstatt zur Erweiterung des Equipments.

App free down home
tipp10.com/de/download/ | tipp10
lathanda.de/index.php/downloads/file/2-eos2-robot | EOS2
atom.io | Atom html editor opensource

Link
matheprisma.de | Binärer Baum u.a.

color-hex.com | html Farben Hexadezimal-code

OOP Java/ Processing | openprocessing

Processing ist eine objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklungsumgebung. Die Programmiersprache ist auf die Einsatzbereiche Grafik, Simulation und Animation spezialisiert. Processing wird in einem quelloffenen Projekt entwickelt, das am Massachusetts Institute of Technology in Boston von Ben Fry (Broad Institute) und Casey Reas (UCLA Design|Media Arts) initiiert wurde. Processing hat den Charakter einer stark vereinfachten Version der Programmiersprache Java, ermöglicht Interaktionen und visuelle Elemente zu programmieren und richtet sich vorwiegend an Gestalter, Künstler und Programmieranfänger.

processing.org | OOP (Objektorientierte Programmierung) mit Java/ Processing. Vereinfachte Java-Version
processing.org/examples | Beispiele
hello.processing.org/editor | Tutorial

py.processing.org/reference | Python/ Processing

openprocessing.org
openprocessing.org/browse/#

python | OOP - Programmierpraxis Extra | mit py computer science circles (Uni Waterloo)

18 Kapitel zur Objektorientierten Programmierung mit Python anhand von Programmier-Beispielen in deutsch und Englisch.

opt Referat eines der Kapitel:
Vortrag auf deutsch mit Verwendung cscircles.cemc.uwaterloo.ca/de
Präsentation in Englisch cscircles.cemc.uwaterloo.ca
Übungen mit Codierung direkt ausführen und kommentieren.

- vgl. cscircles.cemc.uwaterloo.ca > cheatsheet.pdf

OOP Referate (**Python** | Java) (Lit. z.B. python4kids.net > Wie ein Informatiker denken lernen)

Kap02 **Variablen, Ausdrücke und Anweisungen**
Kap03 **Funktionen**
Kap04 **Verzweigung und Rekursion**
Kap05 **Funktionen mit Wert**
Kap06 **Iteration**
Kap07 **Strings**
Kap08 **Listen**
Kap09 **Tupel**
Kap10 **Datentyp Dictionary**
Kap11 **Dateien und Ausnahmen**
Kap12 **Klassen und Objekte**
Kap13 **Klassen und Funktionen**
Kap14 **Klassen und Methoden**
Kap15 **Mengen von Objekten**
Kap16 **Vererbung**
Kap17 **Verkettete Listen**
Kap18 **Stacks**
Kap19 **Queues**
Kap20 **Bäume**

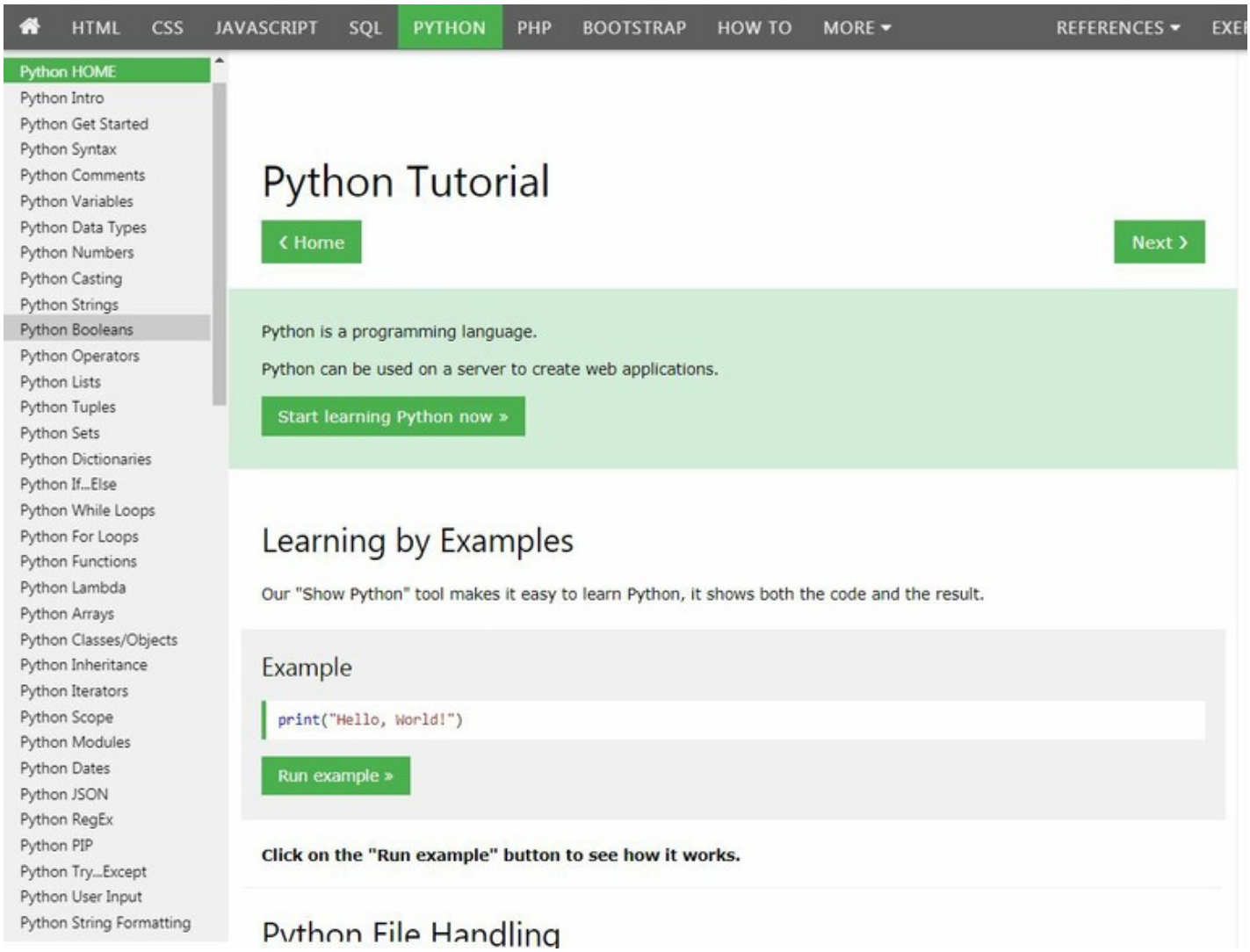
mit objektorientierter Programmierung z.B. turtle Grafik (Python)

#Codierung aufschlussreich kommentieren.
#Eigene Projektdokumentation erstellen

python | Programmierpraxis extra

Recherche z.B. in Online-Quellen, Lit:
Python für kids: http://python4kids.net/how2think/index.html

python | Programmierpraxis extra | Literatur in Englisch | w3schools.com



w3schools.com/python | py
und
w3schools.com/python/python_examples.asp | py examples
w3schools.com/python/exercise.asp | py exercise
w3schools.com/quiztest/quiztest.asp?qtest=PYTHON | py Quiz
w3schools.com/python/showpython.asp?filename=demo_default | py run example

Referate Medieninformatik I (5 min Vortrag, opt. im Team)

Zahlsysteme: Dezimal, Binär, Hexadezimal
Funktionsweise eines Rechners/ Rechnerarchitektur (Von-Neumann): Prozessor (Rechenwerk, Steuerwerk, Arbeitsspeicher, Ein- und Ausgabeeinheiten, Hintergrundspeicher, Datenbus, Adressbus und Steuerbus
Massenspeicher (von Terra- bis Peta- u. Exabyte-Bereich)
UML
Generalisierung und Spezialisierung
Programmiersprachen

'creative coding' - interaktive Grafik | Processing
Computergrafik 2D
digitale Typographie, (web-)Fonts
digitale Photographie, Post-Produktion Ph/ Video (Colourgrading)
3D-Modellierung, Objekt-Design

HTML5, XML, CSS
JavaScript, jQuery
API (MPI, CUDA, Map-Reduce)
Multimedia
webRadio
webTV

ebooks
Mediengestaltung
UX-Design
User-Interface (Komponenten, Navigation)
Screen-Design und Storyboard in WebDesign (CMS)
Interaktive Grafik in CMS-Templates (persona.co, cargo.site)

Sketching with Hardware
Arduino Boards
RaspberryPi
audio-Maschinen, UpDesign audio
FashTech
Mikroprozessoren, Sensoren, Digital- Analog-Wandlung
Simulation in Physik, Elektronik, Mechanik

Betriebssysteme
Datenbanken DB (relationale und noSQL)
Performanz und Laufzeit
Software-Entwicklungstechniken (Agil, Phasen 'Wasserfall": Analyse, Entwurf, Implementierung, Test, Bewertung und Abnahme)
Modell-View-Controller Konzept (MVC)
Software-Muster Kompositum (Composite Pattern)
Mensch-Maschine-Interaktion, KI

Iteration
Rekursion. Rekursive Datenstruktur Liste und Warteschlange
Spezialfälle der verketteten Liste - die Datenstruktur Stapel (Stack) und Schlange (Queue) (LIFO, FIFO)
Binäre Bäume, Baum als spezieller Graph

OSI-Schichtenmodell
Schichtenmodell: Internet Protokollstapel (TCP/IP-Stack)
Rechnernetze, Netzwerk-Topologien
Rechnernetze und GPS, Echtzeit, Mobilfunk, Steuerung
Internet als Kombination von Rechnernetzen
Cloud Computing
Computerarchitekturen (Superskalar, SIMD, Multi- u. Many-Core, Compute-Farm, GPU, Cell, Grid Computing)
Mobile und Verteilte Systeme
Big Data und Data Science
Forensic Architecture
Geo-Mapping (Heatmaps)

It-Sicherheit, Netzwerksicherheit
Kryptographie (Brute-Force-Verfahren,. Laufzeit von Algorithmen und Grenzen der Berechenbarkeit)

Protokolle zur Datenübertragung
Protokolle zur Beschreibung der Kommunikation von Prozessen
Modellierung einfacher, nebenläufiger Prozesse, z.B. mithilfe eines Sequenzdiagramms; Möglichkeit der Verklemmung
Formale Sprachen (Alphabet, Terminalsymbole, Nichtterminalsymbole, leeres Wort; EBNF - Erweiterte Backus-Naur-Form)
Syntaktischer Aufbau einer Sprache: Grammatik (Nichtterminale, Terminale, Regeln, Startsymbol)
Erkennender endlicher Automat zur Syntaxprüfung regulärer Sprachen

Registermaschine als Modell eines Daten verarbeitenden Systems (Datenregister, Befehlsregister, Befehlszähler, Statusregister. Arbeitsspeicher für Programme und Daten, Adressierung der Speicherzellen
Maschinensprache
Zustandsübergänge der Registermaschine als Wirkung von Befehlen
Systemnahe Programmierung und Umsetzung aller Kontrollstrukturen (z.B. mit MiniMaschine)

Compiler (Parsing, Zwischencode, Optimierung, Code-Generierung)
Debugger
Programm-Analyse und Typsysteme (Lambda-Kalkül, Polymorphie, Subtyping, Lineare- u. abhängige Typen, abstrakte Interpretation, Alias- u. Heapanalyse)
Semantic Web Rule Logic
Deklarative Programmiersprachen (Prolog, Datalog, OPS5)
Prolog als deklarative Programmiersprache (Trennung von Arbeits- u. Steuerungs-Algorithmus) + Logik

Multimediale Lehr- und Lernsysteme
Lerntheorien (Behaviorismus, Kognitivismus, Konstruktivismus, soziales Lernen, kooperatives Lernen)

Kreativität
Motivationstheorie

Games (Multiplayerspiele, Wegewahl)
GWAPs - Games with a Purpose, Serious Games

Wissenschaftliches Arbeiten, Recherche-Strategien, Zitieren

LPplus ISB Inf11 Lernbereich 1: Generalisierung (ca. 8 Std.)

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- analysieren und ordnen zweckmäßig hierarchische Strukturen aus ihrer Erfahrungswelt (z. B. Klassifizierung von Tieren) und erstellen entsprechende Generalisierungshierarchien in Form von Klassenmodellen.
- implementieren mithilfe einer objektorientierten Sprache Generalisierungshierarchien unter Berücksichtigung von Vererbung; dabei verwenden sie auch abstrakte Klassen.
- nutzen zur flexiblen Anpassung verschiedener Verhaltensweisen an den jeweiligen Kontext der Anwendungssituation (z. B. bei der rollenabhängigen Berechnung des Gehalts der Mitarbeiter in einem Unternehmen) zielführend das Konzept der Polymorphie durch Überschreiben von Methoden in Unterklassen.

Inhalte zu den Kompetenzen:

- Generalisierungshierarchie: Ober- und Unterklasse, grafische Darstellung der hierarchischen Klassenstruktur
- Generalisierung und Spezialisierung als unterschiedliche Sichtweisen auf dieselbe Klassenbeziehung, Vererbung von Attributen und Methoden auf Unterklassen
- Abstrakte Klasse: Definition und grundlegende Konzeption, abstrakte Methode
- Polymorphismus und Überschreiben von Methoden
- Fachbegriffe: Vererbung, Generalisierung, Spezialisierung, Polymorphismus, Oberklasse, Unterklasse, abstrakte Klasse, abstrakte Methode

LPplus ISB Inf11 Lernbereich 2: Die rekursive Datenstruktur Liste (ca. 27 Std.)

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- modellieren mithilfe einfach verketteter Listen lineare Datenstrukturen aus verschiedenen Situationen ihres Lebensumfeldes (z. B. Warteschlangen, Listen mit Personendaten). Sie nutzen dabei das Softwaremuster Kompositum und erkennen so den Vorteil einer bewährten Modellierungsstrategie.
- entwickeln unter Verwendung des Kompositums Algorithmen für die einfach verkettete Liste, um Elemente hinzuzufügen, zu löschen bzw. Berechnungen über die Listenelemente durchzuführen. Sie nutzen dabei das Prinzip der Rekursion als naheliegende Problemlösungsstrategie.
- implementieren fachgerecht einfach verkettete Listen und die zugehörigen Algorithmen mithilfe einer objektorientierten Programmiersprache.
- bewerten und vergleichen in konkreten Anwendungssituationen dynamische Listenstrukturen mit der statischen Struktur Feld und schärfen damit ihr Bewusstsein für einen zielgerichteten Einsatz der Datenstrukturen.
- nutzen bei der Bearbeitung von Anwendungssituationen aus der Praxis durch fachgerechte Anpassung an die konkrete Aufgabenstellung die durch Trennung von Struktur und Inhalt bedingte universelle Einsetzbarkeit verketteter Listen.

Inhalte zu den Kompetenzen:

- Liste als dynamische Datenstruktur zur Verwaltung von Datenbeständen mit flexibler Anzahl von Elementen versus Feld als statische Datenstruktur.
- Rekursion, rekursive Abläufe: rekursiver Aufruf, Abbruchbedingung, Aufrufsequenz
- einfach verkettete Liste: allgemeines Prinzip, rekursive Struktur, ausgewählte und soweit möglich rekursiv definierte Methoden (u. a. zum Einfügen, Entfernen und Suchen von Elementen sowie zur Bestimmung der Listenlänge)
- Trennung von Struktur und Daten/Inhalt

- Kompositum (Composite Pattern) als Beispiel eines Softwaremusters
- Grundprinzip von Stapel (LIFO) und Warteschlange (FIFO) als Spezialfälle der verketteten Liste
- Fachbegriffe: statische Datenstruktur, dynamische Datenstruktur, (einfach verkettete) Liste, Rekursion, rekursive Methode, rekursiver Aufruf, Abbruchbedingung, Aufrufsequenz, Kompositum, LIFO, FIFO, Softwaremuster

LPplus ISB Inf11 Lernbereich 3: Die rekursive Datenstruktur Baum (ca. 15 Std.)

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- modellieren unter Berücksichtigung des Softwaremusters Kompositum und des Prinzips der Trennung von Struktur und Daten geordnete Binärbäume zu verschiedenen Problemstellungen ihres Erfahrungsbereiches (z. B. digitales Wörterbuch), in denen eine effiziente Datenhaltung wichtig ist. Durch den erneuten Einsatz des Kompositums erkennen sie die universelle Verwendbarkeit von Softwaremustern.
- entwickeln rekursive Algorithmen zur Verwaltung der Daten, die in einem Binärbaum abgespeichert sind (insbesondere zur Traversierung eines Binärbaums sowie zum Einfügen und Suchen von Elementen in einem geordneten Binärbaum), und wenden diese Algorithmen an konkreten Beispielen an.
- implementieren fachgerecht auf Grundlage gegebener Modelle geordnete Binärbäume mithilfe einer objektorientierten Programmiersprache.
- bewerten und vergleichen geordnete Binärbäume mit verketteten Listen hinsichtlich der Effizienz bei Suchanfragen. Ihnen wird damit bewusst, dass insbesondere ein ausbalancierter geordneter Binärbaum eine in Hinblick auf die Suche sehr effiziente Datenstruktur ist.
- nutzen bei der Bearbeitung von verschiedenen Anwendungssituationen aus der Praxis (z. B. Speicherung unterschiedlicher Daten wie Lexikoneinträge oder Kundeninformationen in Binärbäumen) eine bereits implementierte Version eines geordneten Binärbaums und passen diese fachgerecht an die konkrete Aufgabenstellung an. Sie vertiefen dabei ihr Verständnis, dass insbesondere durch das Konzept der Trennung von Struktur und Daten grundsätzlich eine Wiederverwendbarkeit der bereits vorliegenden Implementierung möglich ist.

Inhalte zu den Kompetenzen:

- Baum: Wurzel, Knoten, Kante, Blatt, Pfad, Höhe, Ebene; Binärbaum, Eigenschaften von Binärbäumen: vollständig, balanciert, entartet
- geordneter Binärbaum: Grundkonzept, Einfügen und Suchen von Elementen
- Traversierungsstrategien, d. h. Verfahren zur Auflistung aller Elemente eines Binärbaums: Präorder, Inorder, Postorder
- Fachbegriffe: Höhe, Ebene, Binärbaum (vollständig, balanciert, entartet, geordnet), Traversierung, Präorder, Inorder, Postorder

LPplus ISB Inf11 Lernbereich 4: Die Datenstruktur Graph (ca. 13 Std.)

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- modellieren sachgerecht vernetzte Strukturen (Graphen) im Rahmen praktischer Fragestellungen, z. B. zur Planung von Verkehrsrouten. Dadurch gewinnen sie einen nachhaltigen Einblick in die umfassende Rolle, die Graphen in vielen Bereichen des Alltags spielen.
- klassifizieren Graphen allgemein und an konkreten Beispielen anhand ihrer Eigenschaften.
- implementieren mithilfe einer objektorientierten Programmiersprache und unter Verwendung einer Adjazenzmatrix auf fachgerechte Weise die Datenstruktur Graph.
- erläutern allgemein und an konkreten Beispielen die Idee der Tiefensuche, formulieren den zugehörigen Algorithmus und wenden diesen an konkreten Beispielen an.
- beurteilen die Einsetzbarkeit der Tiefensuche hinsichtlich vorgegebener Anforderungen (z. B. Erreichbarkeit sämtlicher Knoten in einem Graphen, kürzester Weg zwischen zwei Knoten).
- implementieren die Tiefensuche und modifizieren den Algorithmus in geeigneter, vom Anwendungskontext abhängiger Weise (z. B. bei der Auswahl aller Knoten mit bestimmten Eigenschaften).

Inhalte zu den Kompetenzen:

- Eigenschaften von Graphen: gerichtet, ungerichtet, zusammenhängend, unzusammenhängend, bewertet (gewichtet), unbewertet, mit Zyklen, zyklenfrei, Erreichbarkeit von Knoten
- Adjazenzmatrix, zweidimensionales Feld
- Algorithmus zum Graphendurchlauf am Beispiel der Tiefensuche
- Fachbegriffe: gerichtet, ungerichtet, zusammenhängend, unzusammenhängend, bewertet (gewichtet), unbewertet, mit Zyklen, zyklenfrei, Erreichbarkeit (von Knoten), zweidimensionales Feld, Adjazenzmatrix, Tiefensuche

LPplus ISB Inf11 Lernbereich 5: Softwaretechnik – Praktische Softwareentwicklung (ca. 21 Std.)

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- erläutern den Ablauf eines Softwareentwicklungsprojekts anhand der typischen Phasen des Wasserfallmodells.
- planen, strukturieren und koordinieren die Durchführung eines Softwareprojekts zu einer umfangreichen Aufgabenstellung aus der Praxis (z. B. Software zur Inventarverwaltung oder für einen einfachen Routenplaner), indem sie sich an einem etablierten Vorgehensmodell der Softwareentwicklung (z. B. Wasserfallmodell) orientieren. Sie erhalten so einen realistischen Einblick in eine bewährte Vorgehensweise bei der Durchführung komplexer Projekte, wie sie beispielsweise im Berufsalltag auftreten.
- führen das Softwareprojekt entsprechend ihrer Planung im Team durch und berücksichtigen dabei Grundideen bewährter Softwarearchitekturen, wie z. B. Model-View-Controller (MVC). Sie setzen in diesem Zusammenhang geeignete Modellierungstechniken der Informatik (z. B. Klassen-, Zustandsdiagramme) situationsgerecht ein und implementieren den Systementwurf, ggf. unter Nutzung passender rekursiver dynamischer Datenstrukturen und geeigneter Programmbibliotheken.
- prüfen und bewerten im laufenden Entwicklungsprozess mithilfe von praktischen Tests zur frühzeitigen Fehlererkennung die Richtigkeit der Softwarekomponenten hinsichtlich der in der Planung erstellten Spezifikation.
- erstellen eine fachgerechte Dokumentation des Softwareprojekts und präsentieren die Ergebnisse der Projektarbeit in geeigneter Weise.

Inhalte zu den Kompetenzen:

- Grundlagen der Projektplanung: Zielsetzung, Arbeitsteilung, Schnittstellen, Meilensteine, Lasten- und Pflichtenheft
- Wasserfallmodell als klassisches Beispiel eines Vorgehensmodells in der Softwareentwicklung mit den typischen Phasen: Analyse, Entwurf, Implementierung, Test, Bewertung und Abnahme
- Grundkonzept des Softwaremusters Model-View-Controller (MVC)
- Verwendung von Frameworks oder Bibliotheken, z. B. zur Nutzung einer Datenbank oder von Dateien zur persistenten Datenspeicherung
- Fachbegriffe: Vorgehensmodell, Wasserfallmodell, Phasen, Meilenstein, Lastenheft, Pflichtenheft, Softwaremuster Model-View-Controller (MVC)