

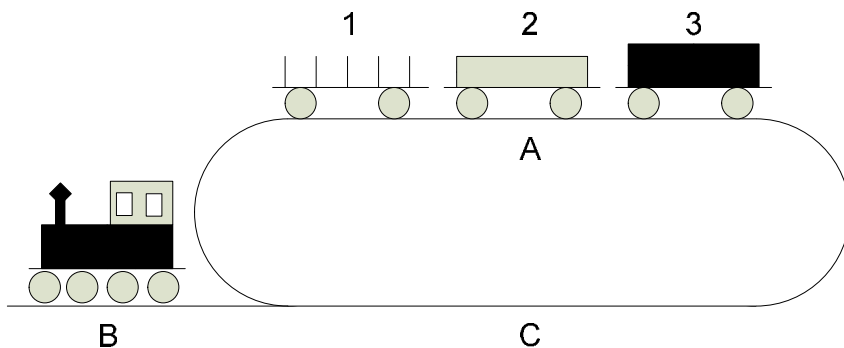
Übungsblätter zum Part 1

Grundlagen der Programmierung



Übung Spezifikation

Spezifikation



Informelle Spezifikation – Rangierproblem:

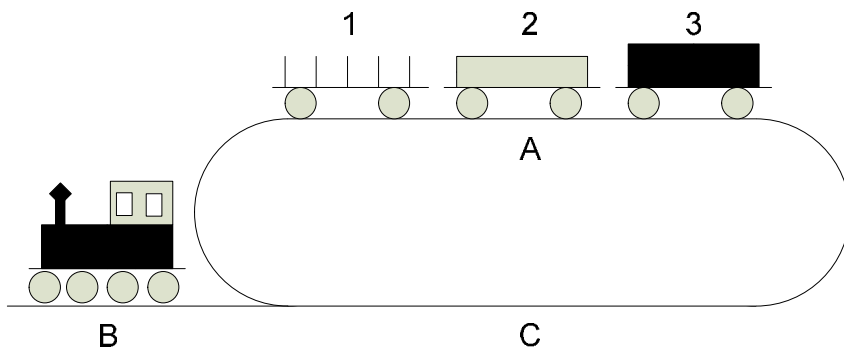
„Eine Lokomotive soll die im Gleisabschnitt A befindlichen Wagen 1, 2, 3 in der Reihenfolge 3, 1, 2 auf Gleisstück C abstellen“

- Bewerten Sie die Vollständigkeit, Detailliertheit und Unzweideutigkeit die aufgezeigten Problembeschreibung!
- Einzelüberlegung 5 min.
- Gemeinsame Diskussion



Quelle: Gumm, H. P.; Sommer, M.: Einführung in die Informatik, 8. Auflage. München, Oldenbourg Verlag, 2009

Spezifikation



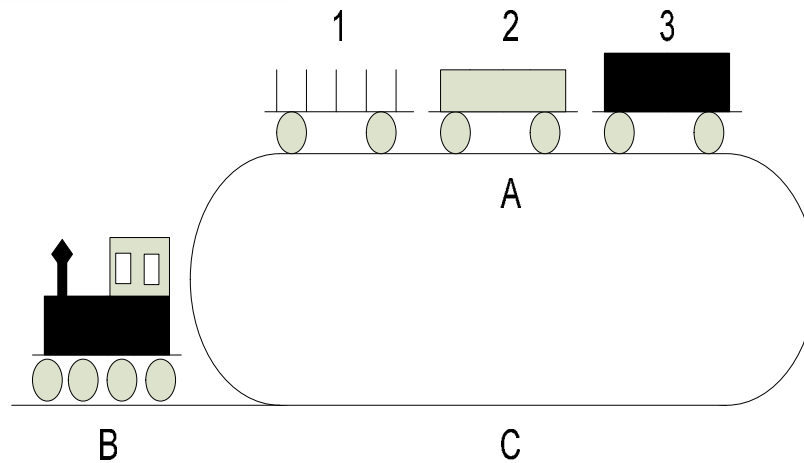
Informelle Spezifikation – Rangierproblem:

„Eine Lokomotive soll die im Gleisabschnitt A befindlichen Wagen 1, 2, 3 in der Reihenfolge 3, 1, 2 auf Gleisstück C abstellen“

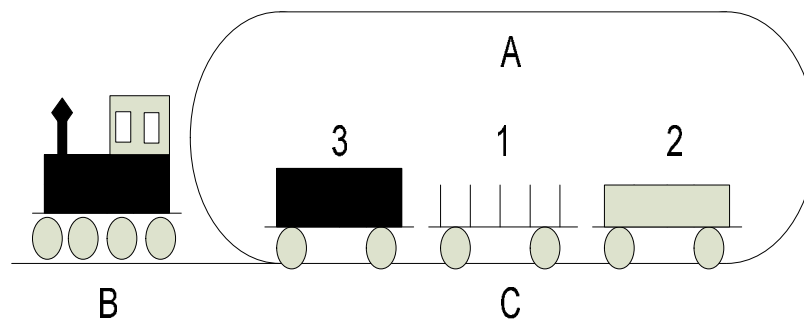
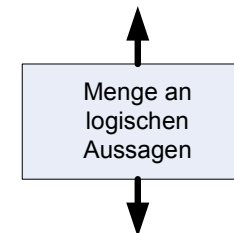
- **Vollständigkeit:**
 - Wie viele Wagen kann die Lokomotive auf einmal ziehen?
 - Wie viele Wagen passen auf das Gleisstück B?
- **Detailliertheit:**
 - Welche Aktionen kann die Lokomotive ausführen (fahren – vor, fahren - zurück, koppeln, entkoppeln, ...)?
- **Unzweideutigkeit:**
 - Darf die Lokomotive am Ende zwischen den Wagen stehen?

Quelle: Gumm, H. P.; Sommer, M.: Einführung in die Informatik, 8. Auflage. München, Oldenbourg Verlag, 2009

Spezifikation



Vorbedingung {P}



Nachbedingung {Q}

Quelle: Gumm, H. P.; Sommer, M.: Einführung in die Informatik, 8. Auflage. München, Oldenbourg Verlag, 2009

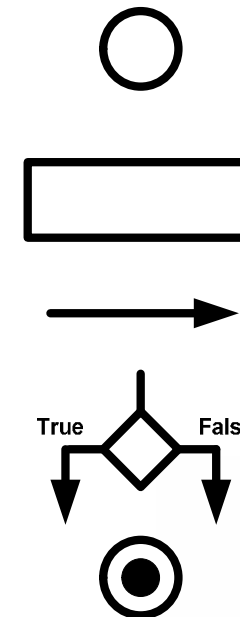


Übung Algorithmen

Algorithmen

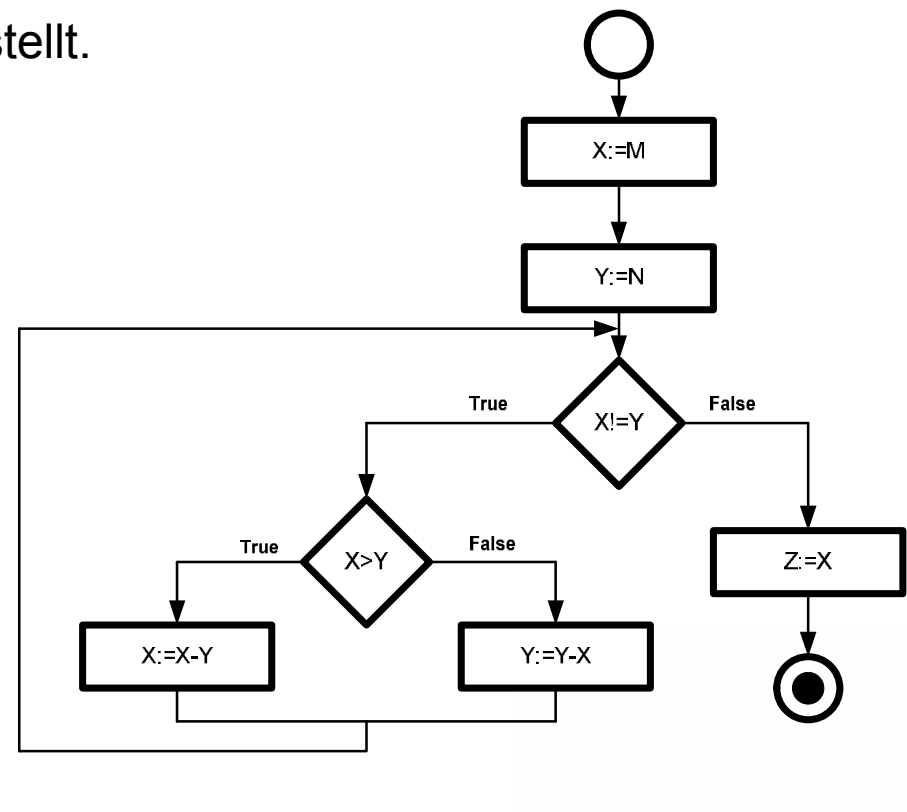


- Entwickeln Sie mit Hilfe der grafischen Notation für Flussdiagramme einen Algorithmus zur Lösung des ggT-Problems.
- Einsatz der folgenden Notationselemente:
 - Beginn der Ausführung - Kreis
 - Elementaraktion – Rechteck
 - Sequenz des Algorithmus - Pfeile
 - Bedingte Verzweigung – True/False
 - Ende des Algorithmus – gefüllter Kreis



Algorithmen

- Als Elementaraktionen werden Wertzuweisungen (z.B. $x := M$) und Berechnungen (z.B. $Y := Y - X$) unterstellt.
- Bemerkung:
 - Nicht direkt auf Rechner abbildbar
 - Zweideutige Notationselemente
 - Sprung oder Schleife
 - Eingesetzte Zeichen wie „:=“
 - Schnell Unübersichtlich



Übung Algorithmen



- Wann kann man einen Algorithmus als korrekt bezeichnen? Wie könnte man diese Eigenschaft nachweisen?
- In welchem Bezug stehen Algorithmen und Programme zueinander? Entscheiden Sie für die folgenden Bereiche, ob es sich um einen Algorithmus oder ein Programm (oder um keines von beiden) handelt:
 - Ergebnis einer Routenplaneranfrage im WWW
 - Wahlprogramm einer Partei
 - Backanleitung für den Brotbackautomat
 - Verhalten eines kundenfreundlichen Fahrscheinautomaten
 - Musiknoten auf einem Notenblatt
- Warum ist der als Beispiel gegebene Algorithmus (ggf) deterministisch?



Übung Algorithmen

- Korrekt – d.h. entspricht der fachlichen Aufgabenstellung, z.B. Berechnung der Wasseroberfläche der Erde, der Nachweis der Korrektheit kann durch nutzerbezogene Tests erfolgen.
- Algorithmen (Mensch) und Programme (Maschine/Rechner)
- Jedes Programm enthält zugleich Algorithmen, umkehrt ist das selbstverständlich nicht so.
- Programme sind eingeschränkter und müssen den Befehlsvorrat der Maschine exakt berücksichtigen.
- Algorithmen können im Fall der generischen Beschreibung leicht auf verschiedene Programmiersprachen des selben Paradigmas übertragen werden.



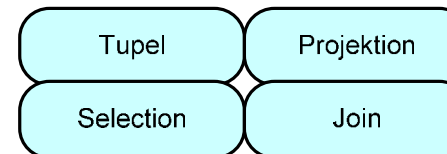
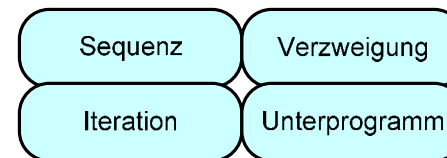
Übung Algorithmen

- Bestimmung Algorithmus A oder Programm P
 - Ergebnis einer Routenplaneranfrage im WWW → A
 - Wahlprogramm einer Partei → kein A oder P
 - Backanleitung für den Brotbackautomat → A
 - Verhalten eines kundenfreundlichen Fahrscheinautomaten → A und P
 - Musiknoten auf einem Notenblatt → A
- Warum ist der als Beispiel gegebene Algorithmus (ggt) deterministisch?
 - Für eine konkrete Parametrisierung ist die Abfolge der einzelnen Verarbeitungsschritte eindeutig festgelegt, d.h. der Kontrollfluss wird immer in der selben Weise durchlaufen.

Programmierparadigmen



- Imperatives Programmierparadigma (z.B. BASIC, Pascal, Modula, Java, C/C++, C#, SMALLTALK) → Fokus auf das WIE
 - Strukturierte Programmierung
 - Prozedurale Programmierung
 - Objektorientierte Programmierung
- Deklarative Programmierparadigma (z.B. Haskell, LISP, Prolog, SQL, XML/HTML) → Fokus auf das WAS
 - Logische Programmierung
 - Constraint-Programmierung
 - Funktionale Programmierung
- Vergleichen Sie imperative und deklarative Programmierparadigmen mit Hilfe einer Literatur-/Internetrecherche! (Optionale Aufgabe)



Fokus: SQL Abfragesprache



Programmierparadigmen

- Augsten, S.: Definition „Programmierparadigmen“ - Was ist ein Programmierparadigma?, <https://www.dev-insider.de/was-ist-ein-programmierparadigma-a-864056>, Abruf: Oktober 2020
- SELFHTML – Einstieg in die Programmiertechnik (Programmiertechnik/Programmierparadigma), <https://wiki.selfhtml.org/wiki/Programmiertechnik/Programmierparadigma>, Abruf: Oktober 2020



Übung Syntax und Semantik

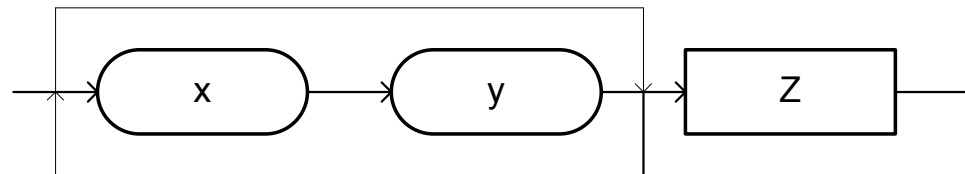
Übung Syntax und Semantik



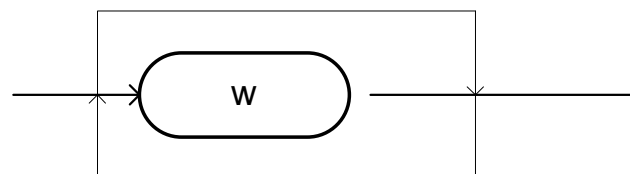
Was ist syntaktisch korrekt, entsprechend der Syntaxdiagramme?

- $xyxyxy$ - **OK**
- $xxxw$ - **FALSCH**
- $xyxyxywww$ - **OK**
- $wwwwwww$ - **OK**
- $xyxyx$ - **FALSCH**
- $wxyxyxy$ - **FALSCH**
- $xyZwww$ - **FALSCH**
- $AxyZ$ - **FALSCH**

Block A



Block Z



5 min. Übung und gemeinsame Auswertung



Übung Syntax und Semantik



- Nennen Sie Ihnen bekannte Regeln zur Syntax von Texten, die in deutscher, englischer oder auch russischer Sprache abgefasst sind?
- Identifizieren Sie im folgenden potentielle Syntaxregeln:
 - Bezeichner (z.B. Variablen) dürfen nicht mit einer Ziffer beginnen.
 - Programme müssen ein korrektes Ergebnis liefern.
 - Jede Programmzeile ist mit einem Fragezeichen abzuschließen.
 - Öffnende und schließende geschweifte Klammern sind immer paarweise bzw. „balanciert“ einzusetzen.
 - Variablen müssen vor ihrer Verwendung deklariert werden.
 - Programme müssen nach endlicher Zeit terminieren.



Übung Syntax und Semantik

- Nennen Sie Ihnen Regeln zur Syntax von Texten, die in deutscher, englischer oder auch russischer Sprache abgefasst sind?
 - Einsetzbarer Zeichenvorrat (z.B. kyrillischer Zeichen).
 - Leserichtung von links nach rechts.
 - Verwendung von Zeilen.
 - Leerzeichen trennen eingesetzte Wörter.
 - Sätze werden mit einem Punkt beendet.



Übung Syntax und Semantik

- Bezeichner (z.B. Variablen) dürfen nicht mit einer Ziffer beginnen. **Syntax**
- Programme müssen ein korrektes Ergebnis liefern. **Semantik**
- Jede Programmzeile ist mit einem Fragezeichen abzuschließen. **Syntax**
- Öffnende und schließende geschweifte Klammern sind immer paarweise bzw. „balanciert“ einzusetzen. **Syntax**
- Variablen müssen vor ihrer Verwendung deklariert werden. **Syntax**
- Programme müssen nach endlicher Zeit terminieren. **Semantik**