

Übung 1: Fragen zu Grundlagen der Datenverarbeitung

1. Was ist ein Programm? Was ist Software?

Was ist ein Programm?

Ein Programm ist die Beschreibung einer Bearbeitungsvorschrift für einen Computer zur Lösung einer vorgegebenen Aufgabe in einer Programmiersprache. Eine Bearbeitungsvorschrift zur Lösung einer Aufgabe wird auch Algorithmus genannt. Ein Programm ist also ein Algorithmus, dessen Anweisungsfolge in einer Programmiersprache formuliert ist.

Was ist Software?

Der Begriff Software ist umfassender als der Begriff Programm. Als Software werden alle immateriellen Teile eines Computers verstanden. Das sind alle Programme, aber auch die zugehörigen Daten und Dokumentationen.

2. Beschreiben Sie das EVA-Prinzip?

Der Datenfluss durch ein Programm erfolgt nach dem so genannten EVA-Prinzip (Eingabe - Verarbeitung - Ausgabe). Programme verarbeiten die Daten, die z. B. vom Benutzer eingegeben werden, und liefern Ergebnisdaten zurück.

- ✓ Dem Programm werden Daten (Informationen) zur Verfügung gestellt (Eingabe).
- ✓ Das Programm verarbeitet diese Daten nach einem vorgegebenen Algorithmus.
- ✓ Die Ergebnisse, die das Programm ermittelt hat, werden ausgegeben.

Übung 2: Fragen zu Programmiersprachen

1. Nach welchen Kriterien können Programmiersprachen klassifiziert werden?

Programmier-sprachen können beispielsweise nach ihrer historischen Entwicklung **nach Generationen** oder gemäß **ihrer Programmiertechnik** klassifiziert werden. Es gibt aber auch andere Kategorisierungsmöglichkeiten.

2. Gehören Assembler-Sprachen zu den höheren Programmiersprachen? Begründen Sie Ihre Antwort.

Assembler-Sprachen gehören **nicht** zu den höheren Programmiersprachen.

Assembler-Sprachen sind wie Maschinensprachen an bestimmte Prozessoren gebunden. Höhere Programmiersprachen sind dadurch gekennzeichnet, dass sie unabhängig von einem bestimmten Computersystem sind.

3. Ordnen Sie Programmiertechniken und Programmiersprachen zu:

Programmier-sprachen	prozedural	objektorientiert	logisch
Lisp			x
C#		x	
C++		x	
Java		x	
Cobol	x		

Übung 3: Modularisierung und Zerlegung in Teilprobleme

Ein Bankkunde kommt zu einem Bankautomat und möchte Geld abheben. Dazu muss er über eine Bankkarte verfügen und sich legitimieren können. Der Vorgang kann in einzelne Teilprobleme zerlegt werden.

Das wäre eine mögliche Zerlegung in Teilprobleme, wobei auch andere Zerlegungen denkbar und legitim sind. Die nachfolgende Zerlegung ist also nur ein Szenario, was eine Abstraktion auf eine ganz bestimmte Situation bedeutet.

1. Identifizieren Sie die möglichen Teilprobleme bzw. Schritte. Beschreiben Sie die Teilprobleme.

Teilproblem 1

Eine Bankkarte muss vorhanden sein und in den Bankautomaten eingeführt werden.

Teilproblem 2

Der Bankkunde muss sich legitimieren. Üblicherweise gibt er dazu eine Geheimzahl ein. Also ist die Eingabe der Geheimzahl das nächste Teilproblem.

Teilproblem 3

Die Geheimzahl wird verifiziert. Sie kann valide oder falsch sein.

Teilproblem 4

Die Geheimzahl ist valide. Dann muss entschieden werden, wieviel Geld ausgezahlt werden soll.

Teilproblem 5

Die Geheimzahl ist nicht valide. Dann muss entschieden werden, ob

- ✓ der Vorgang abgebrochen wird,
- ✓ die Bankkarte eingezogen wird oder
- ✓ ein Bankkunde noch einmal die Geheimzahl eingeben darf.

Typischerweise kann ein Kunde eine Geheimzahl 3x eingeben und beim dritten Versuch wird die Bankkarte eingezogen. Vorher kann jederzeit mit einer Eingabe beim Bankautomaten der gesamte Vorgang abgebrochen werden.

Teilproblem 6

Die Geheimzahl ist valide und es soll Geld ausgezahlt werden soll. Dann muss entschieden werden, ob die Summe möglich ist. Zwei Varianten sollen in diesem Szenario denkbar sein:

- ✓ Das Konto weist die gewünschte Deckung auf und die Summe überschreitet nicht den maximalen Auszahlungsbetrag. Dann erfolgt eine Auszahlung.
- ✓ Das Konto weist nicht die gewünschte Deckung auf oder die Summe überschreitet den maximalen Auszahlungsbetrag. Dann erfolgt keine Auszahlung.

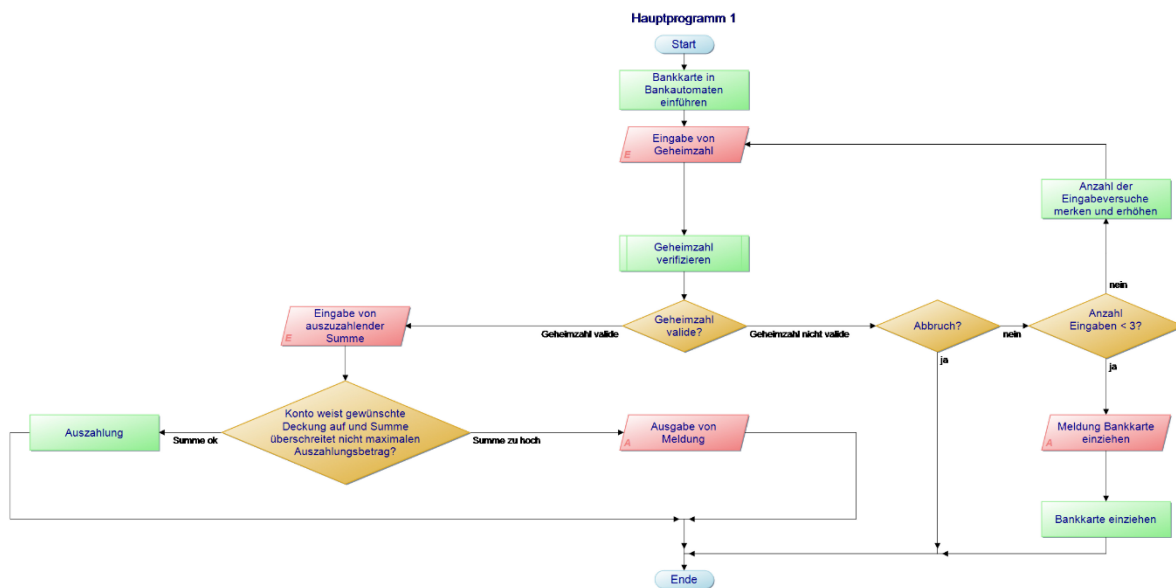
2. Beschreiben Sie die Verbindungen zwischen den Teilproblemen. Wie folgen die einzelnen Schritte aufeinander?

Die Folge der einzelnen Schritte kann in Textform mit einer Aufzählungsstruktur so beschrieben werden:

1. Bankkarte in Bankautomaten einführen
2. Eingabe von Geheimzahl
3. Geheimzahl verifizieren
 - a. Wenn Geheimzahl valide, dann Eingabe von auszahlender Summe.
 - i. Wenn Konto gewünschte Deckung aufweist und Summe überschreitet nicht maximalen Auszahlungsbetrag, dann Auszahlung und Ende.
 - ii. Wenn Konto nicht gewünschte Deckung aufweist oder Summe überschreitet maximalen Auszahlungsbetrag, dann Meldung und Ende.

- b. Wenn Geheimzahl nicht valide, dann Frage nach Abbruch?
- i. Wenn Abbruch, dann Ende.
 - ii. Wenn kein Abbruch, dann Kontrolle, ob die Geheimzahl nicht mehr als 3x eingegeben wurde?
 1. Wenn ja, dann Meldung und Bankkarte einziehen und Ende.
 2. Wenn nein, Anzahl der Eingabeversuche merken und erhöhen und erneute Eingabe von Geheimzahl

Um die Folge der einzelnen Schritte darzustellen, kann man verschiedene Darstellungsmittel verwenden, die so einen Programmablauf viel eindeutiger beschreiben können. Unter anderem einen Programmablaufplan (PAP), was im folgenden Kapitel genauer erläutert, aber sich auch ohne die genaue Kenntnis der Symbole leicht nachvollziehen lässt.



3. Wie könnten die identifizierten Module der Teilprobleme in anderen Vorgängen verwendet werden?

Die Zerlegung in Teilprobleme führt dazu, dass unterschiedliche Standardmodule umgesetzt werden können, die dann auch in anderen Applikationen zum Einsatz kommen könnten. Etwa Folgende:

- ✓ Verifizierungsmodul für die Gültigkeit einer Geheimzahl.
- ✓ Modul zur Überprüfung der gewünschten Auszahlungssumme.
- ✓ Modul zum Einziehen einer Bankkarte.
- ✓ Modul zum Merken der Eingabeversuche einer Geheimzahl.

Übung 4: Fehler in einem Algorithmus finden und verbessern

Ein Programm soll alle geraden Zahlen zwischen 0 und 100 auf dem Bildschirm ausgeben. Dazu soll folgender Algorithmus in Pseudocode verwendet werden, der eine Schleife simulieren soll.

Beginn des Algorithmus

Teste, ob der Wert der Zahl, die in der Variablen i steht, gerade ist

Wenn ja, dann Ausgabe Wert der Zahl i

Wenn nein, dann keine Ausgabe

Gehe wieder zum Test, solange die Variable i nicht den Wert 100 erreicht

Ende des Algorithmus

1. Identifizieren Sie die möglichen Fehler.

- ✓ Die Variable i wurde in dem Algorithmus nicht eingeführt. Der Test, ob der Wert von i gerade ist, kann also nicht funktionieren, da die Variable nicht da ist.
- ✓ Der Wert der Variablen i wird nicht erhöht. Damit kann die Bedingung für das Ende der Wiederholungen nie erfüllt sein. Man erhält eine sogenannte Endlosschleife. Zudem wird immer die gleiche Zahl ausgegeben.

2. Korrigieren Sie die Fehler.

So könnten die Fehler korrigiert werden:

Beginn des Algorithmus

Deklaration der Variable i

Vorbelegung mit dem Wert 0

Teste, ob der Wert der Zahl, die in der Variablen i steht, gerade ist

Wenn ja, dann Ausgabe Wert der Zahl i

Wenn nein, dann keine Ausgabe

Erhöhe den Wert von i um den Wert 1

Gehe wieder zum Test, solange die Variable i nicht den Wert 100 erreicht

Ende des Algorithmus