

5.4 Anwenderkenntnisse

Anwendung von Tabellenkalkulations-Software (Excel, Calc, ...) inkl. Formeln/Funktionen

Eine Tabellenkalkulation ist eine Software für die interaktive Eingabe und Verarbeitung von numerischen und alphanumerischen Daten in Form einer Tabelle. Vielfach erlaubt sie zusätzlich die grafische Darstellung der Ergebnisse in verschiedenen Anzeigeformen.

- Das Bildschirmfenster der Software ist dabei in Zeilen und Spalten eingeteilt.
- Je nach Programm bzw. Bedienungskonzept heißt dieser Bereich zum Beispiel Arbeitsblatt, Worksheet oder Spreadsheet.
- Jede Zelle der Tabelle kann eine Konstante (Zahl, Text, Datum, Uhrzeit ...) oder eine Formel enthalten.
- Für die Formeln stehen meist zahlreiche Bibliotheksfunktionen zur Verfügung.
- Die Formeln können Werte aus anderen Zellen benutzen.
- Bei Änderung der referenzierten Zellen einer Formel aktualisiert die Software den erst angezeigten Wert der Formelzelle normalerweise automatisch, ggf. aber auch nur auf Anforderung.
- Werden Formeln eines Tabellenfeldes an andere Stellen kopiert, muss zwischen absolutem und relativem Zellbezug unterschieden werden.
- Formelzellen können auf andere Formelzellen verweisen.
- Mit diesem Prinzip können komplizierte Rechengänge mit vielen verknüpften Teil-Ergebnissen übersichtlich dargestellt werden.

Anwendung von Textverarbeitungs-Software (Word, Writer, ...) inkl. Formatiermöglichkeiten

Ein Textverarbeitungsprogramm ist ein Computerprogramm zum Verfassen von Textdokumenten. Der erstellte Text kann als Datei abgespeichert und/oder ausgedruckt werden. Der Übergang zu anderen Programmen ist fließend, da sich in vielen Textverarbeitungsprogrammen heute auch Grafiken und Tabellen einbinden lassen.

Mit Word lassen sich alle Arbeiten erledigen, die im weitesten Sinne mit Textbearbeitung zusammenhängen. Das Programm gliedert sich in zehn sogenannte Registerblätter darunter „Start“, „Einfügen“ und „Überprüfen“. Das sind die wichtigsten Wordfunktionen der zentralen acht Registerblätter, die sich horizontal oberhalb vom Text über den gesamten Bildschirm erstrecken:

- Datei: Word-Dokument speichern oder drucken.
- Start: Text produzieren und in Schriftgröße, -farbe, -art verändern.
- Einfügen: Bilder und Tabellen einfügen und formatieren.
- Entwurf und Layout: Design verfeinern.
- Referenz: Inhaltsverzeichnis und Fußnoten ergänzen.
- Sendungen: Serienbriefe verfassen.
- Überprüfen: mit Kommentarfunktion und Korrekturmodus arbeiten.

Folgende Alternativen gibt es zu Microsoft Office Word:

- Google Docs.
- Microsoft Office Word Online.
- OpenOffice.

- LibreOffice.
- WPS Office Free.
- AbiWord.
- SoftMaker FreeOffice.

Word bietet zahlreiche Möglichkeiten verschiedene Texte zu formatieren. Die meisten Formatierungen fallen unter Kategorien wie Text-, Absatz-, und Seitenformatierung an. Zusätzliche Formatierungen, die angewendet werden können, sind Direkte oder Manuelle Anwendungen von Formatierungen.

Ein Dateiformat gibt an, wie Daten für eine bestimmte Anwendung gespeichert werden. Eine DOCX-Datei ist beispielsweise ein Microsoft Word-Dokument im Open-XML-Format.

Anwendung von Bildbearbeitungs-Software

Die Software ist für die Veränderung von Fotos, Negativen, Dias oder digitalen Bildern.

Wird angewandt, um Fehler zu beseitigen, die beim Fotografieren oder anderen Bilderfassungen entstehen können.

Dazu gehören beispielsweise Über- und Unterbelichtung, Unschärfe, Kontrastschwäche, Bildrauschen, Rote-Augen-Effekt und stürzende Linien.

Dadurch wirken Bilder oft zu dunkel, zu hell, zu unscharf oder anderweitig mangelhaft.

Die Ursachen können technische Probleme oder mindere Qualität der Aufnahmegерäte (Digitalkamera, Objektiv, Scanner) sein.

Software:

Kostenpflichtige und kostenlose Programme:

Paintshop Pro, Photoshop Elements, ACDSee Photo Studio, Lightroom, Photoshop, usw.

GIMP, Paint.NET, Photoscape X, Darktable, LightZone, RawTherapee, Apple Fotos

Kenntnis der Unterschiede von offenen, proprietären und plattformunabhängigen Dateiformaten

Offene Dateiformate:

Sind Formate, die ohne Zahlung von Gebühren in beliebigen Programmen verwendet werden können.

z.B.: PDF, HTML, JPEG, PNG, OGG, WEBM, XML, BMP, TXT

Proprietäre Dateiformate:

Sind Formate die eingeschränkt oder nur durch eine Software vom Hersteller verwendet werden können.

Technische und Rechtliche Hindernisse erschweren die Nutzung dieser Dateiformate.

Gegen Zahlung auch durch Programme von Drittfirmen verwendbar.

Vorteil: Kunden sind abhängig von der Hersteller-Software, können nicht einfach wechseln

Nachteil: Wechsel zu einem anderen Software-Hersteller fast unmöglich, z.B.

Microsoft Office (docx), Android (apk), Windows (wma), Audacity (aup), Photoshop (psd)

Plattformunabhängige Dateiformate:

Genauer als plattformübergreifend und allgemeiner portabel - wird in der Informationstechnik die Eigenschaft genannt, dass ein Programm auf verschiedenen Plattformen ausgeführt werden kann.

Webanwendungen - welche vom Browser ausgeführt werden.

Hybrid-Apps - Anwendungen, die unabhängig von der Plattform auf unterschiedlichen Betriebssystemen und unterschiedlichen mobilen Endgerät ausgeführt werden können.

Multi-Channel-Apps - Anwendungen, die unabhängig von Endgerät und Betriebssystem ausgeführt werden können. Diese Geräte können sowohl mobil als auch, im Unterschied zu den Hybrid-Apps, stationär sein.

In Zwischencode vorliegende Software - Programme, die entweder in Form von Bytecode, wie hauptsächlich Java-Programme, oder eines portablen, interpretierbaren Quellcodes (Python, Perl, PHP und andere) vorliegen.

Fat Binaries und Universal Binaries - Programmpakete, die mehrere lauffähige Versionen enthalten. Das Betriebssystem startet ohne Zutun des Anwenders die passende Version.

Quellcode-Portabilität - Diese Form der Plattformunabhängigkeit ist häufig bei C-Programmen für Unix anzutreffen: Der Quellcode enthält Anweisungen, die es erlauben, die Betriebssystemunterschiede auszugleichen.

5.5 Programmiersprachen

Kenntnisse über gängige Programmiersprachen und deren Anwendungsmöglichkeiten

Wichtige Programmiersprachen zusammengefasst, wofür welche Sprache verwendet wird!

Python:

Python wurde von Guido van Rossum entwickelt, um Menschen Code schreiben zu lassen, der so einfach zu verstehen ist wie Englisch. Sie ist eine moderne und objektorientierte Programmiersprache, die plattformunabhängig auf fast allen Betriebssystemen läuft.

Besonders eignet sie sich für einfache und kleine Programme sowie für große Entwicklungen aufgrund seiner Objektorientierung. Sie kann als »Rapid Applikation Development«-Tool (RAD) ebenso eingesetzt werden wie als Scriptsprache zur Systemverwaltung.

Künstliche Intelligenz kann außerdem auch mit Python geschrieben werden, denn auch funktionale Programmierkonstrukte sind in der Sprache integriert. Ein großer Vorteil ist, dass Python eine freie Software ist, die kostenlos erhältlich ist.

Viele ProgrammiererInnen bezeichnen Python als das Schweizermesser unter den Programmiersprachen, weil es in mehreren IT-Bereichen verwendet werden kann: Datenanalyse, System Administration, Web Development, Bioinformatics, Automatisierung, AI, Maschine Learning, Microcontrollers und mehr.

JavaScript:

JavaScript ist die wohl bekannteste Programmiersprache für Webanwendungen. Damit lassen sich nicht nur dynamische Webseiten programmieren, sogar Spiele und richtige Anwendungen sind möglich.

Das Gute an JavaScript ist, dass es im Browser ausgeführt werden kann und damit unabhängig ist. Weiterhin hat sich Node.js unter den Server-Seitigen Sprachen einen Namen gemacht als auch Frameworks wie React Native, die dem Entwickler erlauben mit JavaScript-Code mobile Apps zu entwickeln. Somit ist JavaScript ein wirklicher Allrounder.

Die Sprache ist ein absoluter Favorit für jeden, der viele verschiedene Computerprogramme und Apps für Android und iOS entwerfen möchte.

Java:

Diese Programmiersprache hat sich schon auf vielen Millionen Geräten etabliert und ist plattformunabhängig. Das bedeutet, sie ist für Windows, Linux und Mac OSX gleichermaßen geeignet.

Was die Sprache so besonders macht ist ihre vielfältige Einsatzweise. Sie kann für Anwendungen, Apps und Spiele verwendet werden und beinhaltet immer neueste technische Konzepte der Softwareentwicklung.

Mit Java können vielseitige Programme geschrieben werden, jedoch bietet sie sich besser für größere Projekte an als für kleine Anwendungen oder Prototypen. Ein großer Einsatzbereich ist beispielsweise die Server-seitige Programmierung von Web-Anwendungen.

PHP:

PHP ist eine Programmiersprache, die zur Webseitenprogrammierung verwendet wird. Hauptaufgabe der Sprache ist das Ausspielen einer Webseite und vor allem, was im Hintergrund auf dem Webserver, dem sogenannten Backend passiert.

Durch PHP kann man beeinflussen, welcher Inhalt und Teil der Webseite dem Nutzer angezeigt werden soll. Hierbei wird das Programm immer auf dem Webserver ausgeführt und gestartet, auf dem die Webseite ansässig ist.

Besonders beliebt ist hier die Abfrage von Datenbanken, um bestimmte Daten auf einer Webseite anzuzeigen. in Kombination mit JavaScript sowie HTML & CSS ist PHP eine sehr einflussreiche Sprache.

SQL:

SQL ist keine direkte Programmiersprache, da mit ihr keine Programme geschrieben werden. SQL steht für "Structured Query Language" und wird zur Kommunikation zu Datenbanken verwendet. Hiermit kann man diese erstellen, füllen und filtern.

Sie wird außerdem oft in Verknüpfung mit Webseiten benutzt, um z.B. einen Artikel im Webshop zu speichern. Um SQL zu nutzen, macht es Sinn sich im Vorfeld mit Datenbanken auseinanderzusetzen, um deren Funktion und Wirkungsweise zu verstehen.

HTML & CSS:

Die beiden Sprachen, die eigentlich keine Programmiersprachen sind, werden gerne im Zusammenhang mit "richtigen" Programmiersprachen erwähnt, da sie wichtig für die Erstellung und Gestaltung von Webseiten sind.

Mit HTML und CSS (Cascading Style Sheets) beschreibt man die Inhalte, die Struktur und das Design von Webseiten, die nachher im Browser ersichtlich sind. Mit HTML und CSS lässt es sich als Anfänger am einfachsten anfangen, da die Gestaltung des Frontend einen leichten Einstieg in die Logik und den Aufbau der Programmierung bietet.

Unterschied prozedurale und objektorientierte Programmierung

Prozedurale Programmierung und Objektorientierte Programmierung (OOP) sind zwei Programmierparadigmen.

Ein Programmierparadigma ist ein grundlegender Stil beziehungsweise Herangehensweise der Computerprogrammierung. Diese unterscheiden sich darin, wie verschiedene Elemente des Programms konzeptionell dargestellt und umgesetzt werden, aber auch im Hinblick darauf, wie die zum Lösen von Problemen erforderlichen Schritte definiert sind.

Prozedurales Programmieren ist eine Art des Programmierens bei der angestrebt wird, ein bestimmtes Problem zu lösen, indem man die Menge der dazu benötigten Schritte und deren Abfolge bestimmt, um das gewünschte Ergebnis oder den gewünschten Zustand zu erreichen. Eines der Hauptkonzepte der prozeduralen Programmierung ist der Prozeduraufruf. Eine Prozedur, die auch als Subroutine, Methode oder Funktion bezeichnet wird, enthält eine geordnete Liste von auszuführenden Anweisungen. Eine Prozedur kann jederzeit während der Ausführung durch eine andere Prozedur oder für sich selbst aufgerufen werden. Beispiele für typische prozedurale Programmiersprachen sind C und Pascal.

Programme bestehen oft aus Modulen, die Teile eines Programms sind, die separat codiert und getestet werden können und dann zu einem vollständigen Programm zusammengesetzt werden. In prozeduralen Sprachen (etwa C) sind diese Module Prozeduren, wobei eine Prozedur eine Abfolge von Anweisungen ist. In C zum Beispiel sind Prozeduren eine Folge von imperativen Anweisungen, wie Zuweisungen, Tests, Schleifen und Aufrufe von Unterprozeduren. Diese Prozeduren sind Funktionen, die Argumente entgegennehmen und letztendlich abhängige Rückgabewerte liefern.

Objektorientierte Programmierung ist ein Programmierparadigma, das in erster Linie auf Abstraktion (in Form von Klassen und Objekten) zurückgreift, um auf der realen Welt basierende Modelle zu erstellen. Eine objektorientierte Anwendung verwendet eine Anzahl von Objekten, die kommunizieren, indem sie Nachrichten an entsprechende Funktionen zur Annahme solcher übergeben. Objekte können Nachrichten weiterleiten, Nachrichten empfangen und Daten verarbeiten. Ziel der objektorientierten

Programmierung ist es, die Flexibilität und die Wartbarkeit von Programmen zu erhöhen.

Da Programme, die mit einer OO-Sprache erstellt wurden, modular sind, können sie einfacher verteilt entwickelt und nach der Entwicklung einfacher verstanden werden.

Fachbegriff Implementierung

Implementierung steht synonym zu den Begriffen Durchführung, Umsetzung, Einführung, Ausführung und Realisierung und bezieht sich im Kontext von itwissen.info gleichermaßen auf Software, Hardware, Protokolle, Netzwerke, Systeme und Programme.

Bei einer Implementierung kann es sich um das Einfügen eines neu entwickelten Systems oder einer Software in ein vorhandenes Netzwerk, System oder Programm handeln. Um die Umsetzung vorgegebenen Prozessabläufen, wobei die Zielvorgaben und Rahmenbedingungen berücksichtigt werden müssen. Es kann sich auf einen Protokoll Stack beziehen, in den ein spezielles Protokoll eingefügt wird, oder auf ein virtuelles Netz, in dem logische Workgroups realisiert werden, es kann sich auf unterschiedliche Protokollausführungen beziehen oder auf Zeichensätze in die anwenderspezifischen Zeichen eingefügt werden.

Die Implementierung erfolgt unter dem Betriebssystem in der von den Prozessoren verwendeten Programmiersprache. Auf den Funktionen des Betriebssystemkerns können Module für anwendungsorientierte Dienste implementiert werden. So beispielsweise die virtuelle Speicherverwaltung und andere Verfahren für die logische Steuerung der Peripheriegeräte. Mit einbezogen in die Implementierung sind alle Arbeitsgänge, die das ordnungsgemäße Funktionieren des zusätzlich Eingebachten mit dem bisher Vorhandenen garantieren.

Implementierung ist der Freiraum, den Entwickler für das Einfügen eigener Soft- oder Hardwarekomponenten in ein bestehendes System oder Programm haben. Sie kann eine Ergänzung zu den bestehenden und laufenden Programmen oder Systemen bilden.

Fachbegriff Compiler

Ein Compiler ist ein Computerprogramm, das den gesamten Quellcode eines Software-Projekts noch vor dessen Ausführung in Maschinensprache übersetzt. Erst im Anschluss wird das Projekt dann vom Prozessor ausgeführt, dem dadurch von Beginn an sämtliche Anweisungen in Maschinencode zur Verfügung stehen. Auf diese Weise hat der Prozessor alle notwendigen Bausteine parat, um die jeweilige Software auszuführen, Eingaben zu verarbeiten und die Ausgabe zu erzeugen. In vielen Fällen findet während des Compilierungsprozesses aber noch ein entscheidender Zwischenschritt statt: Vor der endgültigen Übersetzung in Maschinensprache wandeln die meisten Compiler Quellcode nämlich häufig zunächst in einen Zwischencode (auch „Objektcode“) um, der oft für verschiedene Plattformen geeignet ist und zudem von einem Interpreter verwendet werden kann.

Compiler legen im Rahmen der Code-Generierung fest, welche Anweisungen dem Prozessor in welcher Reihenfolge übermittelt werden. Sofern die Instruktionen nicht voneinander abhängig sind, kann dieser die Anweisungen dabei sogar parallel verarbeiten.

Fachbegriff Interpreter

Ein Interpreter ist ein Computerprogramm, das den Quellcode eines Software-Projekts während dessen Laufzeit – also während es ausgeführt wird – verarbeitet und als Schnittstelle zwischen diesem Projekt und dem Prozessor fungiert. Dabei

geht ein Interpreter immer Codezeile für Codezeile vor, sodass die einzelnen Anweisungen der Reihe nach gelesen, analysiert und für den Prozessor aufbereitet werden. Dieses Prinzip gilt auch für wiederkehrende Anweisungen, die jeweils neu ausgeführt werden, sobald sie an der Reihe sind. Für die Verarbeitung des Software-Codes greifen Interpreter auf eigene, interne Bibliotheken zurück: Ist eine Quellcode-Zeile in die entsprechenden, maschinenlesbaren Befehle umgewandelt, wird sie direkt an den Prozessor weitergeleitet.

Der Umwandlungsprozess ist erst dann abgeschlossen, wenn der gesamte Code interpretiert ist. Er wird lediglich dann vorzeitig unterbrochen, wenn bei der Verarbeitung ein Fehler auftritt – ein Umstand, der die Fehlerbehandlung erheblich vereinfacht, da die problematische Codezeile sofort mit dem Auftauchen des Fehlers gefunden ist.

Hinweis:

Zu den bekanntesten Programmiersprachen, die bei der Umwandlung von Quellcode in Maschinencode überwiegend auf einen Interpreter setzen, zählen BASIC, Perl, Python, Ruby und PHP. Häufig fasst man diese Sprachen daher auch unter dem Begriff „Interpreter-Sprachen“ zusammen.