

BLAISE PASCAL MAGAZINE 109

Multi platform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



Das Semantische Web wird bestimmt kommen BlockChain
Bild zu Text API

Die Abrundungsfunktion in FastReport von CHAT.gpt.

Ein ganzzahliger Quadratwurzel-Algorithmus
Faktoren und Faktorisierung
Ein ganzzahliger Kubikwurzel-Algorithmus

Pseudozufallszahlen und echte Zufallszahlen
*Quantencomputer wurden jetzt eingesetzt,
um direkt mit normalen Maschinen kombiniert für echte Zufallszahlen*

Indizierung Ihrer PDF-Dateien
für eine schnelle Suche innerhalb von Seiten / des gesamten Dokuments

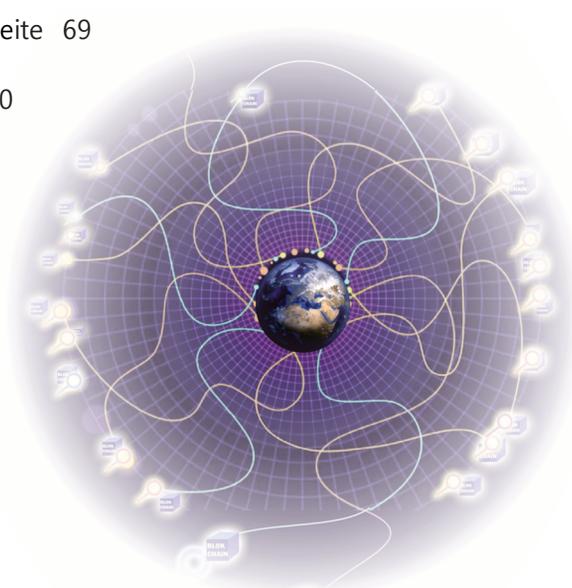
Debugging in Lazarus/FPC



INHALT

ARTIKEL

- Von Ihrem Redakteur Seite 4
 Von unserem technischen Berater - Humor durch Jerry King Seite 5
 Das Semantische Web wir bestimmt kommen Seite 18
Von Detlef Overbeek
- BlockChain Seite 20
Von Detlef Overbeek
- Bild zu Text API Seite 11
Von Max Kleiner
- Die Abrundungsfunktion in FastReport von CHAT.gpt. Seite 69
Von Detlef Overbeek
- Ein ganzzahliger Quadratwurzel-Algorithmus Seite 30
Von David Dirkse
- Faktoren und Faktorisierung Seite 33
Von David Dirkse
- Ein ganzzahliger Kubikwurzel-Algorithmus Seite 36
Von David Dirkse
- Pseudozufallszahlen und echte Zufallszahlen: Seite 6
*- Quantencomputer wurden jetzt eingesetzt, um
 direkt mit normalen Maschinen kombiniert
 für echte Zufallszahlen*
Von Detlef Overbeek
- Indizierung Ihrer PDF-Dateien für eine schnelle Suche
innerhalb von Seiten / des gesamten Dokuments Seite 48
Von Michael van Canneyt
- Debugging in Lazarus/FPC Seite 40
Von Martin Friebe



WEB3
 Web 3.0 wird das Internet in eine stärker reglementierte
 Version verwandeln?
 Die Idee einer dezentralisierten Version des Internets,
 die normalen Nutzern mehr Macht geben würde, stößt
 auf reges Interesse. Außerdem gibt es die Möglichkeit
 eines Semantischen Webs.

WERBUNG

- | | |
|---|-----------------|
| Fastreport Ankündigung | Seite 67 |
| Components for Developers | Seite 74 |
| Database Workbench | Seite 16 |
| Fast Report | Seite 68 |
| Help for Ukraine | Seite 73 |
| Lazarus Handbook Pocket | Seite 39 |
| Lazarus Handbook PDF + Subscription | Seite 17 |
| PDF Viewer 2023 Blaise Pascal Library USB stick | Seite 8/9/28/29 |
| Subscription 2 year | Seite 47 |
| Delphi Community Edition | Seite 66 |



Niklaus Wirth

Pascal ist eine imperative und prozedurale Programmiersprache, die Niklaus Wirth (links unten) in den Jahren 1968-69 entworfen und 1970 veröffentlicht hat. Es handelt sich dabei um eine kleine, effiziente Sprache, die gute Programmierpraktiken unter Verwendung von strukturierter Programmierung und Datenstrukturierung fördern soll. Ein Derivat, bekannt als Object Pascal, wurde 1985 für die objektorientierte Programmierung entwickelt. Der Name der Sprache wurde zu Ehren des Mathematikers und Erfinders der ersten Rechenmaschine gewählt: Blaise Pascal (siehe oben rechts).

Herausgeber: © Foundation Supporting Programming Language Pascal - mit Sitz in den Niederlanden
 Registrierter Name: Stichting Ondersteuning Programmeertaal Pascal IJsselstein, Netherlands
 VAT / BTW: NL814254147B01 Chamber of commerce (KVK) 30 202429 Handy: +31 6 21 23 62 68



CONTRIBUTORS

Stephen Ball http://delphiaball.co.uk DelphiABall	Dmitry Boyarintsev dmitry.living @ gmail.com	Michael Van Canneyt .michael @ freepascal.org	Marco Cantù www.marcocantu.com marco.cantu @ gmail.com
David Dirkse www.davdata.nl mail: David @ davdata.nl	Benno Evers b.evers @ everscustomtechnology.nl	Bruno Fierens www.tmssoftware.com bruno.fierens @ tmssoftware.com	Holger Flick holger @ flixments.com
Mattias Gärtnernc- gaertnma@netcologne.de	Max Kleiner www.softwareschule.ch max @ kleiner.com	John Kuiper john_kuiper @ kpnmail.nl	Wagner R. Landgraf wagner @ tmssoftware.com
Vsevolod Leonov vsevolod.leonov@mail.ru	Andrea Magni www.andreamagni.eu andrea. magni @ gmail.com www.andreamagni.eu/wp		
		Paul Nauta PLM Solution Architect CyberNautics paul.nauta @ cybernautics.nl	
Kim Madsen www.component4developers.com kbmMW		Boian Mitov mitov @ mitov.com	
	Jeremy North jeremy.north @ gmail.com	Detlef Overbeek - Editor in Chief www.blaisepascal.eu editor @ blaisepascal.eu	
Anton Vogelaar ajv @ vogelaar-electronics.com	Danny Wind dwind @ delphicompany.nl	Jos Wegman Corrector / Analyst	Siegfried Zuhr siegfried @ zuhr.nl

Chefredakteur

Detlef D. Overbeek, Niederlande Tel.: Mobil: +31 (0)6 21.23.62.68

Nachrichten und Pressemitteilungen nur per E-Mail an editor@blaisepascal.eu

Abonnemente können online unter <https://www.blaisepascalmagazine.eu/deutsche-Ausgabe/> oder per schriftlicher Bestellung abgeschlossen werden oder indem Sie eine E-Mail an office@blaisepascal.eu senden. Das Abonnement kann zu einem beliebigen Zeitpunkt beginnen. Alle Ausgaben, die im Kalenderjahr des Abonnements veröffentlicht werden, werden geschickt. Das Abonnement hat eine Laufzeit von 365 Tagen. Abonnemente werden nicht ohne Vorankündigung verlängert

Der Zahlungseingang wird per E-Mail verschickt. Sie können das Abonnement bezahlen, indem Sie die Zahlung an folgende Adresse senden:

ABN AMRO Bank Konto Nr. 44 19 60 863 oder per Kreditkarte oder Paypal Name: Stiftung Pro Pascal (Stichting Programmeertaal Pascal)

IBAN: NL82 ABNA 0441960863 BIC: ABNANL2A Umsatzsteuer-Nr.: 81 42 54 147 (Stichting Programmeertaal Pascal)

Abonnementsabteilung Edelstenenbaan 21 / 3402 XA IJsselstein, Niederlande Mobil: + 31 (0) 6 21.23.62.68 office@blaisepascal.eu

Markenzeichen Alle verwendeten Markenzeichen sind Eigentum der jeweiligen Inhaber. Vorbehalt Obwohl wir uns bemühen sicherzustellen, dass die in der Zeitschrift veröffentlichten Informationen korrekt sind, können wir keine Verantwortung für Fehler oder Auslassungen übernehmen. Wenn Sie etwas bemerken, das möglicherweise nicht korrekt ist, wenden Sie sich bitte an den Herausgeber, und wir werden gegebenenfalls eine Korrektur veröffentlichen.



Mitglied der **Königlich Niederländischen Bibliothek**

KB

Mitglied und Spender von **WIKIPEDIA**

Subscriptions (2022 prices)

Internat. excl. VAT

Internat. incl. 9% VAT

Shipment

TOTAL

Printed Issue (8 per year)	±60 pages :	€ 200	€ 218	€ 130	€ 348
Electronic Download Issue (8 per year)	±60 pages :	€ 64,20	€ 70		

COPYRIGHT-HINWEIS

Das gesamte in Blaise Pascal veröffentlichte Material unterliegt dem Copyright © SOPP Stichting Ondersteuning Programmeertaal Pascal, sofern nicht anders angegeben, und darf nicht ohne schriftliche Genehmigung kopiert, verbreitet oder neu veröffentlicht werden. Die Autoren erklären sich damit einverstanden, dass der zu ihren Artikeln gehörende Code nach der Veröffentlichung den Abonnenten zur Verfügung gestellt wird, indem er auf der Website der PGG zum Download angeboten wird, und dass Artikel und Code auf verteilbaren Datenträgern gespeichert werden. Die Nutzung von Programmlisten durch Abonnenten zu Forschungs- und Studienzwecken ist erlaubt, jedoch nicht zu kommerziellen Zwecken. Die kommerzielle Nutzung von Programmlistings und Code ist ohne die schriftliche Genehmigung des Autors untersagt.



Von Ihrem Redakteur

Liebe Leser,
endlich ist es Frühling, und das bedeutet für mich Sonnenschein und Freude im Garten.

Aber leider: Ich habe so viel zu schreiben und so viele neue Themen, über die ich Ihnen berichten möchte, dass ich noch keine Zeit dafür hatte.

Für viele meiner Autoren habe ich ein sehr interessantes Buch gekauft, dessen Lektüre ich Ihnen empfehle: *The Mind of a Bee*, von Lars Chittka, erschienen bei press.princeton.edu. Es ist faszinierend, wozu ein so kleines Lebewesen fähig ist. Sie werden überrascht sein.

Was mich am meisten überrascht hat, war das inzwischen allgegenwärtige ChatGPT. Bereits Version 4. Ich finde es ein sehr hilfreiches und inspirierendes Tool. Aber das war's auch schon.

Alles, was ich von den Leuten höre, ist, dass es sehr intelligent ist.

Zunächst einmal ist es nicht intelligent, was auch immer es ist. Es ist ein sehr großer Algorithmus, der anscheinend über ein sehr großes Wissen verfügt. Kein Wunder, wenn man das gesamte Internet zur Verfügung hat.

Und genau da liegt der Fehler: Er ist sehr oft weit von der Realität oder der Wahrheit entfernt.

Es werden sogar Antworten oder Geschichten erfunden. Das ist zumindest amüsant und zeigt, dass es sich um ein von Menschen geschaffenes Spielzeug handelt.

Nichts ist uns fremd. Nicht einmal ChatGPT.

Spielen Sie damit und haben Sie Spaß.

Apropos Spaß: Ich habe gerade mit dem Schreiben für das nächste Buch begonnen, das wir vorstellen wollen: Programmieren lernen für das Web mit PAS2JS. Wir haben bereits 10 Kapitel und vielleicht kommen noch mehr.

In diesem Buch legen wir Wert auf Einfachheit für Anfänger. Es ist für Einsteiger gedacht.

Natürlich müssen Sie ein gewisses Interesse und Kenntnisse in HTML haben: das hilft, aber wir wollen es Ihnen so einfach wie möglich machen, so dass Sie sogar Ihren eigenen Server damit aufbauen können.

Wenn Sie erst einmal den Dreh raus haben, werden Sie sehr ermutigt sein: es macht einfach Spaß, mit PAS2JS und Pascal eine Website zu erstellen.

Sobald das Buch fertig ist, werden wir einige Veranstaltungen durchführen, bei denen Sie in der Praxis lernen können. Ich werde es Sie wissen lassen.

Ich habe ein paar Zeilen über das Web3 und das Semantische Web geschrieben. Es ist umstritten, aber vielleicht sollten wir ihm eine Chance geben: Wenn es das Web sicherer macht, würde ich mich freuen, aber es sollte nicht zu so etwas wie Big Brother werden.

David Dirkse hat wieder einige Artikel verfasst, die für Anfänger in der Mathematik sehr interessant sein könnten.

Ich mag seine Beispiele sehr, und er ist für sie sehr bekannt geworden. Jetzt hat er drei davon für Sie erstellt.

Michael van Canneyt und ich arbeiten sehr hart an dem Lazarus PDF Kit. Es wächst ständig und wird immer besser. Lesen Sie den Artikel und Sie werden verstehen, dass PDF keine einfache Sache ist.

Zu guter Letzt, und sehr interessant, hat Max Kleiner darüber geschrieben, wie man ein Weinflaschenetikett entziffert. Sehr hilfreich, wenn Sie sich mit Wissen betrinken wollen.

Viel Spaß beim Lesen...

Detlef



Von unserem technischen Berater Jerry King



Meine Damen und Herren, da wir in eine Wolke voller Daten fliegen, werden wir eine gewisse Röhrenbildung erleben. Bitte kehren Sie zu Ihren Sitzen zurück und legen Sie Ihre Sicherheitsgurte an.

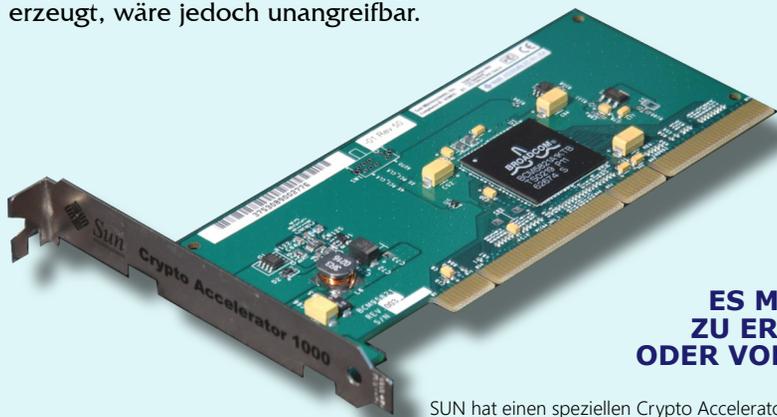


QUANTENCOMPUTER WURDEN NUN ZUM ERSTEN MAL EINGESETZT DIREKT MIT "NORMALEN" MASCHINEN KOMBINIERT

Alle derzeitigen Techniken zur Generierung von Zufallszahlen, die für eine sichere Kommunikation benötigt werden, haben irgendeinen Fehler, aber aktuelle Quantencomputer könnten in der Lage sein, Zufallszahlen zu generieren, die nicht gefälscht werden können.

So enthüllte Edward Snowden 2013, dass die US-Regierung eine Schwachstelle in den Zufallszahlengeneratoren des US National Institute of Standards and Technology eingebaut hatte, die es ermöglichte, die Ausgabe des Generators vorherzusagen.

Alle heute verwendeten Zufallszahlengeneratoren sind pseudozufällig - sie sind so schwer vorherzusagen, dass ihre Ausgabe zufällig erscheint. Aber Angriffe sind denkbar. Ein Zahlengenerator, der wirklich zufällige Ergebnisse erzeugt, wäre jedoch unangreifbar.



**MIT QUANTEN-TECHNIKEN IST
ES MÖGLICH ECHTE ZUFALLSZAHLEN
ZU ERZEUGEN, DIE NICHT GEFÄLSCHT
ODER VORHERGESAGT WERDEN KÖNNEN.**

SUN hat einen speziellen Crypto Accelerator entwickelt.

TRULY RANDOM ODER PSEUDO RANDOM ERKLÄRT

PSEUDO RANDOM

Eine RANDOM NUMBER kann gefälscht werden, wenn die zur Generierung der Zahl verwendete Methode nicht wirklich zufällig ist. Wenn ein vorhersehbarer Algorithmus oder eine vorhersehbare Formel verwendet wird, um eine Zahl zu generieren, erscheint sie zwar zufällig, kann aber von jedem, der den Algorithmus oder die Formel kennt, reproduziert werden. Das bedeutet, dass die Zahl nicht wirklich zufällig ist, sondern eher pseudo-zufällig.

PSEUDO-RANDOM-ZAHLENGENERATOREN

werden häufig in Computerprogrammen verwendet, um Zufallszahlen zu erzeugen. Diese Generatoren verwenden mathematische Formeln oder Algorithmen, um eine Folge von Zahlen zu erzeugen, die zufällig erscheinen, aber tatsächlich deterministisch sind. Da der zur Generierung der Zahlen verwendete Algorithmus bekannt ist, kann jemand die Zahlenfolge reproduzieren, indem er den Algorithmus zurückentwickelt.

Um echte Zufallszahlen zu generieren, ist eine Zufallsquelle erforderlich, z.B. atmosphärisches Rauschen oder radioaktiver Zerfall. Diese Quellen werden verwendet, um einen Startwert für einen Zufallszahlengenerator zu erzeugen, der dann den Startwert verwendet, um eine Folge von Zufallszahlen zu erzeugen. Es ist zwar immer noch möglich, dass jemand die Zahlenfolge vorhersagt, wenn er den Seed* kennt, aber die Unvorhersehbarkeit des Seeds macht dies praktisch unmöglich. **(Ein zufälliger Seed oder Seed-Status oder einfach nur Seed) ist eine Zahl (oder ein Vektor), die zur Initialisierung eines Pseudozufallszahlengenerators verwendet wird).*



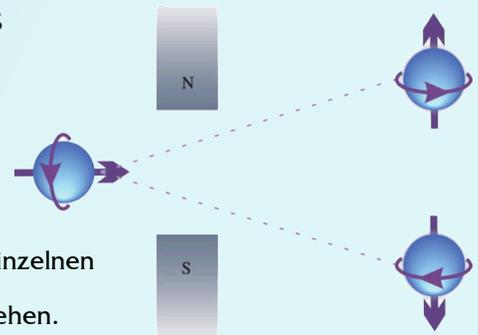
QUANTUM COMPUTERS HAVE NOW BEEN USED FOR THE FIRST TIME TO BE DIRECTLY COMBINED WITH "NORMAL" MACHINES.

WIRKLICH RANDOM

Zusammenfassend lässt sich sagen, dass eine Zufallszahl gefälscht werden kann, wenn sie mit einem Pseudo-Zufallszahlengenerator erzeugt wurde, der einen vorhersehbaren Algorithmus oder eine Formel verwendet. Um echte Zufallszahlen zu erzeugen, ist eine Zufallsquelle erforderlich.

Bei der QUANTENZUFALLSZAHLENGENERATION wird eine Quantenmessung an einem Quantensystem durchgeführt und das Ergebnis zur Erzeugung einer Zufallszahl verwendet. Da Quantenmessungen von Natur aus probabilistisch sind, ist die resultierende Zahl ECHT ZUFÄLLIG und kann nicht vorhergesagt werden, selbst wenn der gesamte Messapparat und die Umgebung bekannt sind.

Diese Methode zur Erzeugung von Zufallszahlen gilt als unknackbar, da sie sich auf die fundamentale Unvorhersehbarkeit der Quantenmechanik stützt. Sie gilt auch als sicherer als andere Methoden zur Generierung von Zufallszahlen, da sie NICHT ANGRIFFEN WERDEN KANN, DER AUF DER KENNTNIS DES ALGORITHMEN ODER DES SEEDWERTES beruht, der in Pseudo-Zufallszahlengeneratoren verwendet wird. Mit Hilfe von Quantentechnologien ist es möglich, echte Zufallszahlen zu erzeugen, die nicht gefälscht oder vorhergesagt werden können.



Aufgrund der Tatsache, dass Quantenprozesse probabilistischen Gesetzen folgen und es daher unmöglich ist, das Ergebnis eines einzelnen Prozesses genau vorherzusagen, haben viele Menschen die Quantenmechanik als Quelle für unbrechbare Zufallszahlen angesehen. Aber das hat einen fundamentalen Fehler: Es gibt keine Möglichkeit, dies doch zu tun.

Wenn Sie nicht vor Ort wären, um den Vorgang zu beobachten, hätten Sie keine Möglichkeit zu wissen, ob eine Zahl echt oder gefälscht ist.

Scott Aaronson und Shih-Han Hung von der University of Texas in Austin haben nun ein Protokoll entwickelt, mit dem echte Zufallszahlen ohne diese Voraussetzung zertifiziert werden können. Dazu fordern sie einen Quantencomputer auf, einen Test in einer bestimmten Zeitspanne zu absolvieren, den nur er bestehen kann. Ein Quantencomputer kann den Test nur bestehen, wenn er die Quantenberechnung ausführt, für die er konzipiert ist, oder eine, die ihr einigermaßen nahe kommt.

Laut Carlos Perez Delgado (*von der University of Kent in Großbritannien*) handelt es sich um eine bemerkenswerte Arbeit und die einzige praktische Möglichkeit, echte Zufälligkeit zu bestätigen, ohne den Prozess physisch beobachten zu müssen.

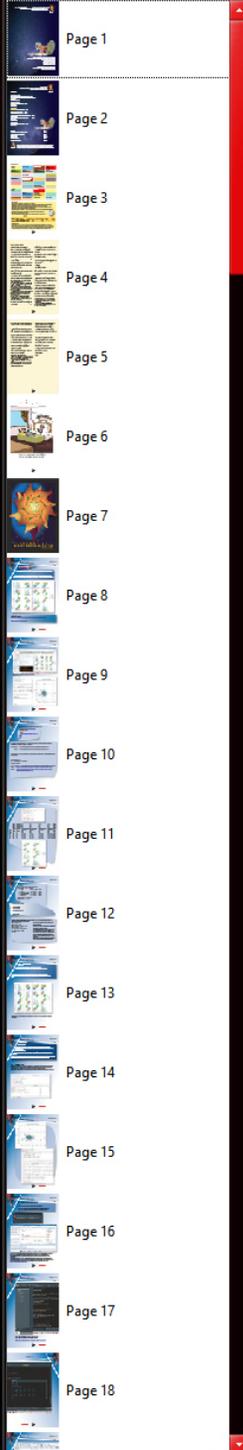
Angesichts der beträchtlichen Rechenressourcen, die für die Überprüfung des Quantenergebnisses erforderlich sind, wird es möglicherweise nicht viele Anwendungen geben, sagt er, da viele Pseudo-Zufallszahlengeneratoren für Cybersicherheitsanwendungen zuverlässig genug und schwer zu knacken sind.

Angesichts der Summe, die auf dem Spiel steht, könnte eine dieser Anwendungen für das ETHERUM-Kryptowährungsnetzwerk sein. Es verwendet Zufallszahlen, um seine Transaktionen zu verifizieren und neues Geld zu schaffen, und ist mehrere hundert Milliarden Dollar wert (im Falle Europas eine Milliarde).

Die neue Strategie wird jedoch in vielen Situationen nicht benötigt. Einige Gegner dieses Protokolls würden lediglich behaupten, dass es übertrieben ist. Für so gut wie jede Anwendung müssen Sie nicht beweisen, dass eine Zahlenfolge zufällig in dem Sinne ist, dass das Universum sie als zufällig betrachtet.

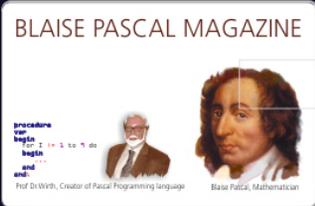
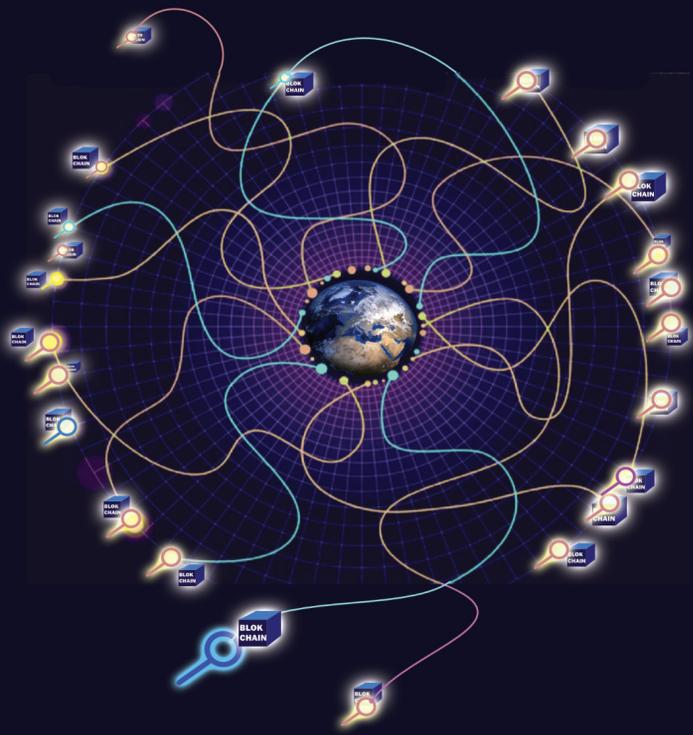


PDF VIEWER 2023 ON CREDIT CARD USB STICK INCLUDING BLAISE PASCAL LIBRARY



BLAISE PASCAL MAGAZINE 109

Multi platform /Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



~~€ 100~~

Special offer € 75

More than 6500 pages
Issue Nr 1 bis 109 all
English spoken

<https://www.blaisepascalmagazine.eu/product/lib-stick/>

SEARCH PDF: BLAISEPASCALMAGAZINE_109_UK.PDF

Load PDF...



140%

indexer

Search in PDF

Click a result to view the corresponding result.



Page 47:
d German • Free Lazarus PDF Kit **indexer** • 8
Issues per year • minimal 6

Page 48:
we showed how to create a PDF **indexer** : a program to analyse a bunch

Page 51:
Search class is part of the fp **indexer** unit presented in the previous

Page 54:
d in the indexed PDFs. The PDF **indexer** stores for each word the name

Page 66:
INCLUDING THE NEW LAZARUS KIT PDF **indexer**



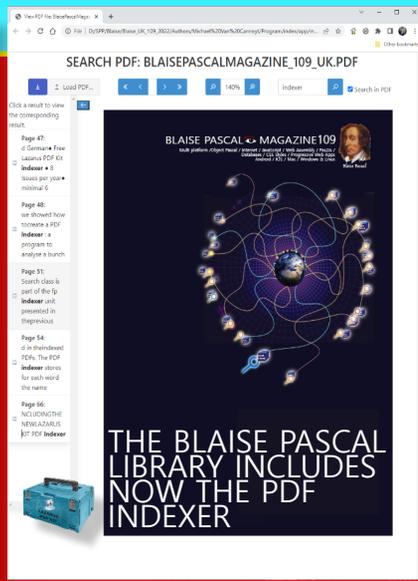
BLAISE PASCAL MAGAZINE 109
Multi platform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js / Databases / CSS Styles / Progressive Web Apps / Android / iOS / Mac / Windows & Linux

THE BLAISE PASCAL LIBRARY (THE LIBSTICK LIBRARY ON USB STICK) INCLUDES NOW THE PDF INDEXER

More than 6500 pages Issue Nr 1 to 109

<https://www.blaisepascalmagazine.eu/product/lib-stick/>

1



2

LAZARUS HANDBOOK



3



4

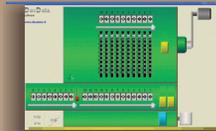
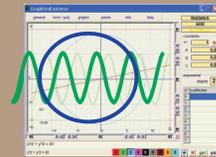


LEARN TO PROGRAM USING LAZARUS

HOWARD PAGE-CLARK

DAVID DIRKSE

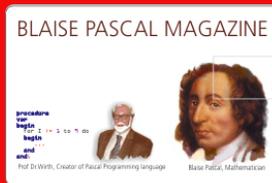
including 50 example projects



BLAISE PASCAL MAGAZINE
COMPUTER (GRAPHICS) MATH & GAMES IN PASCAL

INCLUDING THE NEW LAZARUS KIT PDF INDEXER

1. One year Subscription
2. The newest LIB Stick
 - All issues 1- 109
 - On Credit Card
3. Lazarus Handbook
 - PDF including Code
4. Book Learn To Program
 - using Lazarus PDF including 19 lessons and projects
5. Book Computer Graphics Math & Games
 - PDF including ±50 projects



SUPER
PACK
2023
PRICE € 150
NORMAL PRICE € 275

Bild zu Text API

maXbox Starter 103 - Texterkennung von Zeichen in Bildern



"Ein Bild sagt mehr als tausend Worte. Eine Schnittstelle ist mehr wert als tausend Bilder."

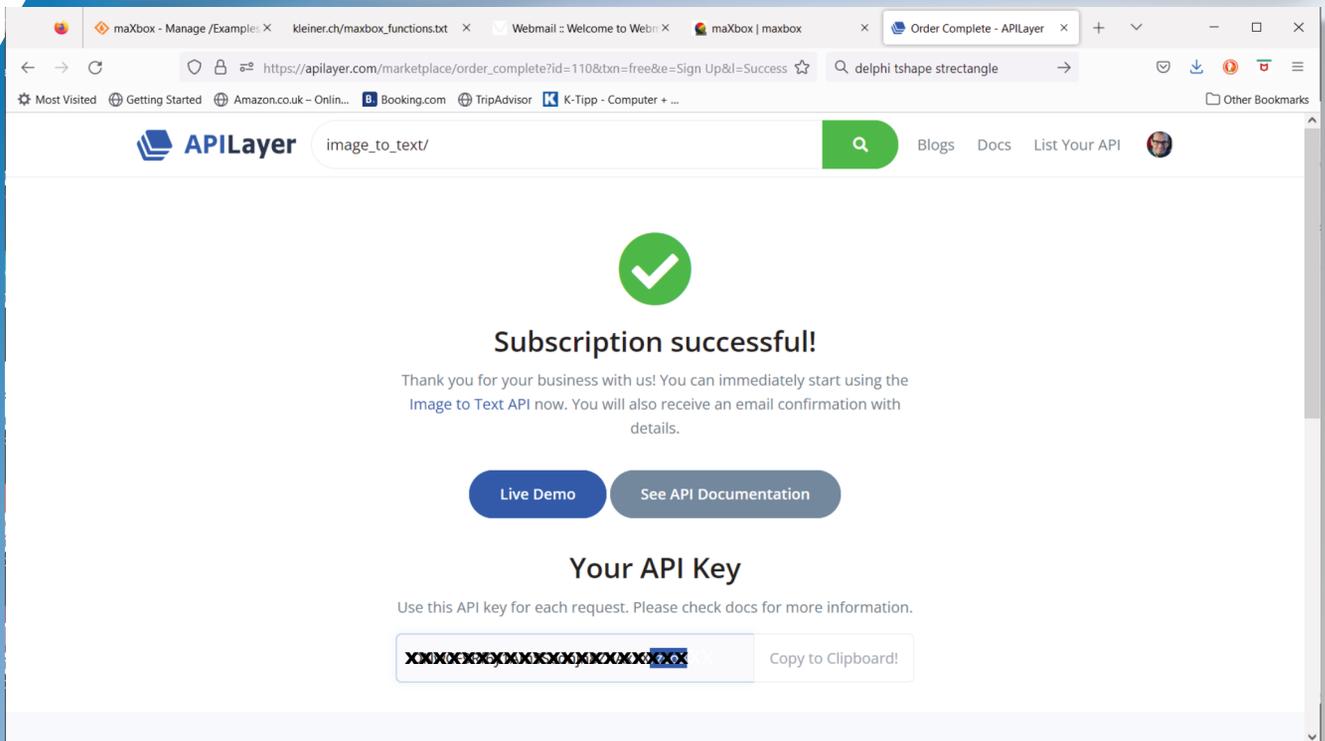
Diese API erkennt und liest einen in Bilder oder Fotos eingebetteten Text.

Image to Text API verwendet eine OCR-Engine auf der Basis eines neuronalen Netzes (LSTM), die sich auf die Zeilen Erkennung konzentriert, aber auch die Erkennung von Zeichenmustern unterstützt.

Sie unterstützt sowohl Handschrift als auch gedruckte Materialien und Straßenkarten.

APILayer ist ein API-Marktplatz, auf dem auch Ihre API ein breiteres Publikum erreichen kann, aber zunächst benötigen Sie einen kostenlosen API-Schlüssel:

Das Ergebnis eines einfachen Abonnements ist der unten stehende Screenshot:



Dieses Register erlaubt Ihnen eine monatliche Nutzung von 30 erfolgreichen Aufrufen.

Für fast alle APIs gibt es ein kostenloses Abonnement. Wenn Sie sich das folgende Buchcover ansehen, können Sie die Textinformationen leicht extrahieren, obwohl das Cover Schatten hat und mit einem Winkel positioniert ist. Wir verwenden WinHttp.WinHttpRequest, JSONObjects und die TGraphics-Bibliothek zum Laden und Testen des REST-Clients. Außerdem übergeben wir den API-Schlüssel als Request-Header, also holen Sie sich zuerst einen Schlüssel bei:

<https://apilayer.com/marketplace>

Sie können auch eine leistungsstarke 'OCR'-Funktion (*Text-in-Bild-Erkennung*) verwenden, um während des Konvertierungsprozesses Text aus einem Bild zu extrahieren. In diesem Fall erhalten Sie als Ergebnis ein bearbeitbares Textdokument, das Sie nach Bedarf anpassen und ändern können. Die dargestellten Daten sind JSON-Daten mit dem gesamten extrahierten Text und sogar die Sprache des zu scannenden Textes wird automatisch erkannt. Bevor wir in den Code eintauchen, hier der Hauptteil des Skripts→:



```
function Image_to_text_API2(AURL, url_imgpath, aAPIkey: string): string;
var httpq: THttpConnectionWinInet;
    rets: TStringStream;
    heads: TStrings; iht:IHttpConnection2;
begin
    httpq:= THttpConnectionWinInet.Create(true);
    rets:= TStringStream.create("");
    heads:= TStringlist.create;
    try
        heads.add('apikey='+aAPIkey);
        iht:= httpq.setHeaders(heads);
        httpq.Get(Format(AURL,[url_imgpath]),rets);
        if httpq.getresponsecode=200 Then result:= rets.datastring
        else result:='Failed: '+
            itoa(Httpq.getresponsecode)+Httpq.GetResponseHeader('message');
    except
        writeln('EWI_HTTP: '+ExceptionToString(exceptiontype,exceptionparam));
    finally
        httpq:= Nil;
        heads.Free;
        rets.Free;
    end;
end;
```

Die Funktion des Hauptteils öffnet eine Verbindung, ruft die API auf und liefert einen Stream, den wir in einen Datenstring umwandeln.

Image2Text oder **Image to Text** Live Demo bietet einen API-Dienst auf seiner APILayer Veröffentlichungsplattform. Die Live-Demo-Funktion ermöglicht es Ihnen, die API in Ihrem Browser zu testen, ohne dass Sie etwas installieren oder programmieren müssen. Sie können alle Parameter nach Belieben ändern und mit der API von vielen Sprachen aus interagieren. Das API-Exportformat ist JSON, z.B. unser Buchcover siehe unten:

The screenshot shows a web browser window with the following content:

- Browser tabs: maXbox - Manage /Example..., kleiner.ch/maxbox_function.txt, Webmail: Welcome to Webm..., maXbox | maxbox, Image to Text API - APILayer
- Address bar: https://apilayer.com/marketplace/image_to_text-api?live_demo=show
- Page title: Image to Text API Playground
- Form: GET /url, url: REQUIRED:
- Code editor: Python code using requests library to call the API with a specific URL and API key.
- Output: JSON response containing the detected text from the image.

Bild zu Text API

maXbox Starter 103 - Texterkennung von Zeichen in Bildern



Die veröffentlichten Ergebnisdatensätze basieren auf LSTM in Kombination mit einer OCR. LSTM steht für Long Short-Term Memory und ist eine Art rekurrentes neuronales Netzwerk (RNN). Bei einem RNN handelt es sich um ein Netzwerk, das die aktuelle Eingabe verarbeitet, indem es die die vorherige Ausgabe (Feedback) berücksichtigt und in seinem Speicher als Speicherzellen für eine kurze Zeitspanne (Kurzzeitgedächtnis). Nehmen wir zum Beispiel unser Buchcover als Eingabe:

LSTMs haben Rückkopplungsverbindungen und Zellen, die unterscheiden sie sich von traditionelleren Feed-Forward neuronalen Netzwerken mit dem immer noch bestehenden Problem des verschwindenden Gradienten Problem.

Diese Eigenschaft ermöglicht es LSTMs, ganze Sequenzen von Daten (z.B. Zeitreihen, Handschriften oder Sätze) zu verarbeiten, ohne jeden Punkt in der Sequenz unabhängig zu behandeln, sondern vielmehr nützliche Informationen über frühere Daten in der Sequenz wie "objektorientiert", "modellieren", "und", "entwickeln" als Kontext. Die Ausgabe des Aufrufs



```
writeln(Image to_text_API2(URL_APILAY,
URLIMAGEPATH4,
DNwCF9Rf6y1AmSSednjn8ZhAxYXr----' ));
```

is the JSON datastring in about Runtime: 0:0:3.859:

```
{"lang": "de", "all text":
"ih \u00bb Der Entwickler\nFachwissen
f\u00fcr Programmierer\
nMax Kleiner\nUML\nmit Delphi\nObjektorientiert modellieren\
nund entwickeln\nSoftware & Support", "annotations": ["ih", "\u00bb",
"Der", "Entwickler", "Fachwissen", "f\u00fcr", "Programmierer", "Max",
"Kleiner", "UML", "mit", "Delphi", "Objektorientiert", "modellieren",
"und", "entwickeln", "Software", "&", "Support"]}
```

Auch das bekannte **Tesseract 4.0** (wie **OmniPage**) hat eine neue OCR-Engine, die auf neuronalen LSTM-Netzen basiert.

Die API kann auch mit diesen wenigen Zeilen P4D-Code ausgelöst werden:

```
procedure PyCode(imgpath: string);
begin
with TPythonEngine.Create(Nil) do begin
pythonhome:= 'C:\Users\max\AppData\Local\Programs\Python\Python36-32\';
try
loadDLL;
ExecString('import requests');
ExecStr('url= "https://api.apilayer.com/image_to_text/url?
url='+imgpath+'");
ExecStr('payload = {}');
ExecStr('headers= {"apikey": "dy5L70eQx72794XBZ8sewEgYTZR85----"}');
Println(EvalStr('requests.request("GET",url, headers=headers,
data=payload).text'));
except
raiseError;
finally
unloadDLL;
free;
end;
end;
end;
```



Bild zu Text API

maXbox Starter 103 - Texterkennung von Zeichen in Bildern



Wenn Sie mit einem eingeschränkten Aufruf oder einem ungültigen Schlüssel scheitern, erhalten Sie eine Reihe von Ausnahmen wie die folgende:
wininet error: Unauthorized (401). or {"message": "Invalid authentication credentials"}

The fact that error code is not "one of the expected return values" tells for the versions that the error comes from an underlying layer and this API just passes it up on internal failure. To shine a bit more light on those errors a function exists to convert the **ErrorCode** to a string for better understanding:

```
function GetWinInetError(ErrorCode: Cardinal): string;
const winetdll = 'wininet.dll';
var Len : Integer; Buffer: PChar;
begin
  Len := FormatMessage(
    FORMAT_MESSAGE_FROM_HMODULE or FORMAT_MESSAGE_FROM_SYSTEM or
    FORMAT_MESSAGE_ALLOCATE_BUFFER or FORMAT_MESSAGE_IGNORE_INSERTS or
    FORMAT_MESSAGE_ARGUMENT_ARRAY,
    Pointer(GetModuleHandle(winetdll)), ErrorCode, 0, @Buffer, SizeOf(Buffer),
    nil);
  try
    while (Len > 0) and {$IFDEF UNICODE}(CharInSet(Buffer[Len - 1], [#0..#32,
    '.', ''])) {$ELSE}(Buffer[Len - 1] in [#0..#32, '.', '']) {$ENDIF} do
      Dec(Len);
      SetString(Result, Buffer, Len);
    finally
      LocalFree(HLOCAL(Buffer));
    end;
  end;
end;
```

Unser letztes Beispiel ist ein interessantes. Wie wäre es mit einer alten Weinflasche (1993 mit Formen (und Trauben ♀)), einem alten historischen Gemälde von 1619, Daten, Symbolen und Positionswinkeln:

https://my6code.files.wordpress.com/2022/12/wine_1993_20221230_141947.jpg?w=768



Und das Ergebnis ist überzeugend, auch die Tatsache, dass die Jahreszahl im Etikettenbild korrekt als 1619 erkannt wurde:
 Vergrößern sie diese aufschrift

```
{ "lang": "it", "all_text":
  "DS\nPODERE CAPACCIA\n1619\nLuggo deffo apacia\nQUERCIAGRANDE\nVino da tavola della Toscana\n1993\nProdotto e imbottigliato all'origine\nPodere Capaccia\nPacini Giampaolo & C. s.a.s.\nRadda in Chianti (SI) - Italia\nItalia",
  "annotations": [ "DS", "PODERE", "CAPACCIA", "1619", "Luggo", "deffo",
    "apacia", "QUERCIAGRANDE", "Vino", "da", "tavola", "della", "Toscana",
    "1993", "Prodotto", "e", "imbottigliato", "all'origine", "Podere", "Capaccia",
    "Pacini", "Giampaolo", "&", "C.", "s.a.s.", "Radda", "in", "Chianti", "(", "SI", ")", "-", "Italia", "Italia"] }
```



Fehlende oder unvollständige Daten sind ohne Kenntnis des Originals nur schwer zu finden. Auf der anderen Seite ist es aber auch sehr einfach zu bedienen, da die Benutzer nur einen Screenshot des Teil, den sie konvertieren möchten, einen Screenshot machen und den Text anschließend kopieren.

Außerdem wird der API-Zugang über eine REST-ähnliche Schnittstelle (Representational State Transfer) bereitgestellt zur Verfügung gestellt, die Datenbankressourcen oder vorab trainierte Modelle im JSON-Format mit Content-Type in der Kopfzeile der Antwort.

HINWEIS: Wenn eine Programmiersprache nicht im Code-Beispiel der Live-Demo aufgeführt ist, können Sie trotzdem API-Aufrufe tätigen, indem Sie eine HTTP-Anforderungsbibliothek verwenden, die in unserer Programmiersprache Sprache geschrieben ist, wie wir es mit GET oder POST getan haben.

```
110 begin
111   httpq:= THttpConnectionWinInet.Create(true);
112   rets:= TStringStream.create('');
113   heads:= TStringlist.create;
114   try
115     heads.add('apikey='+aAPIkey);
116     iht:= httpq.setHeaders(heads);
117     httpq.Get(Format(AURL,[url_imgpath]),rets);
118     if httpq.getResponsecode=200 Then result:= rets.datastring
119     else result:='Failed:'+
120       itoa(Httpq.getResponsecode)+Httpq.GetResponseHeader('message');
121   except
122     writeln('EWI_HTTP: '+ExceptionToString(exceptiontype,exceptionparam));
123   finally
124     httpq:= Nil;
125     heads.Free;
126     rets.Free;
127   end;
128 end;
129
130 procedure PyCode(imgpath: string);
```

Interface List: 1176_APILayer_D

procedure FilelessWB(encodedurl:
procedure FilelessWB2(encodedU
function image_to_text_aPI(AUR
function Image_to_text_API2(AU
procedure PyCode(imgpath: strin
Locs: 160 - code blocks: 5

```
{"lang": "de", "all_text": "ih \u00bb Der Entwickler\nFachwissen f\u00fcr Programmierer\nMax Kleiner\nUML\nmit Delphi\nObjektorientiert modellieren\ntwickeln\nSoftware & Support", "annotations": ["ih", "\u00bb", "Der", "Entwickler",  
"Fachwissen", "f\u00fcr", "Programmierer", "Max", "Kleiner", "UML", "mit", "Delphi", "Objektorientiert", "modellieren", "und",  
"entwickeln", "Software", "&", "Support"]}  
mX4 executed: 12/01/2023 14:50:06 Runtime: 0:0:3.218 Memload: 37% use  
PascalScript maXbox4 - RemObjects & SynEdit
```

SCHLUSSFOLGERUNG:

Die Image to Text API von API Layer erkennt und extrahiert Text aus Bildern mithilfe modernster Algorithmen zur optischen Zeichenerkennung (OCR) in Kombination mit einem neuronalen Netzwerk namens LSTM.

Es kann Texte in verschiedenen Größen, Schriftarten und sogar Handschriften oder schwierige Zahlen erkennen.

Bild zu Text API

maXbox Starter 103 - Texterkennung von Zeichen in Bildern

Referenz:

https://apilayer.com/marketplace/image_to_text-api
<https://apilayer.com/docs>

<https://my6.code.blog/2022/09/02/webpostdata/>
http://www.kleiner.ch/kleiner/images/uml_buch.jpg

Doc und Tool: <https://maxbox4.wordpress.com>

Script Ref: 1176_APILayer_Demo1.txt

Anhang: Eine Delphi REST-Client-API zur Nutzung von REST-Diensten, die in einer beliebigen Programmiersprache geschrieben wurden, mit einer Klasse aus der maXbox4-Integration:
<https://github.com/fabriziocolombo/delphi-rest-client-api/blob/master/src/HttpConnectionWinInet.pas>

Die API ist für Delphi 7 oder höher konzipiert; neuere Versionen nutzen die Vorteile von Generics Methods.

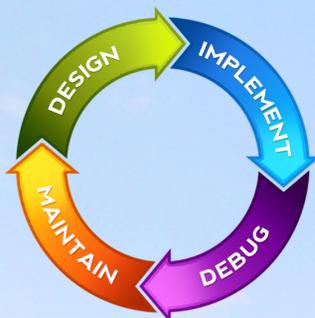
<https://github.com/fabriziocolombo/delphi-rest-client-api>



Introducing

Database Workbench 6

database development environment



Consistent user interface, modern code editors, Unicode enabled, HighDPI aware, ER designer, reverse engineering, meta data browsing, visual object editors, meta data migration, meta data compare, stored routine debugging, SQL plan visualizer, test data generator, meta data printing, data import and export, data pump, Grant Manager, DBA tasks, code snippets, SQL Insight, built in VCS, report editor, database meta data search, numerous productivity tools and much more...

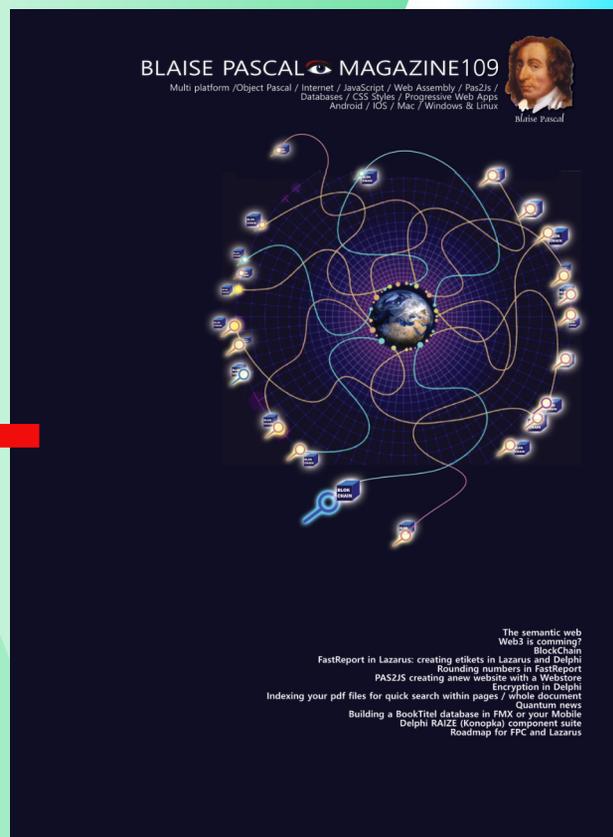
for SQL Server, Oracle, MySQL, MariaDB, Firebird, InterBase, NexusDB and PostgreSQL

 *Database tools for developers*
www.upszene.com

LAZARUS HANDBOOK (POCKET) + SUBSCRIPTION 1 YEAR

- **Lazarus Handbook**
- Printed in black and white
- PDF Index for keywords
- Almost 1000 Pages
- Including 40 Examples
- **Blaise Pascal Magazine**
- English and German
- Free Lazarus PDF Kit Indexer
- 8 Issues per year
- minimal 60 pages
- Including example projects and code

SPECIAL OFFER € 75
+ shipping for the book





By Detlef Overbeek

WEB3

Web 3.0 soll das Internet in eine stärker reglementierte Version verwandeln?

Die Idee einer dezentralisierten Version des Internets, die normalen Nutzern mehr Macht geben würde, stößt auf reges Interesse. Außerdem gibt es die Möglichkeit eines Semantischen Webs.

Es gibt Stimmen, die aus offensichtlichen Gründen eine neue Version des Internets fordern:

- 1 um eine deutlich bessere digitale Zukunft für das Internet
- 2 die Technologie-Giganten, die derzeit unsere Online-Welt (Internet) beherrschen, sind von massiven Missbräuchen erschüttert, sie verlieren eine Menge Geld - durch die Schädigung ihrer Märkte - und scheinen unfähig oder nicht willens zu sein, die Probleme der Online Desinformation, Belästigung und Hass-reden zu lösen.

SOLLTE ES DEZENTRALISIERT SEIN?

Das bedeutet, dass es statt einiger weniger großer Anbieter, die alle registrierten Daten (Provider usw.) überwachen, eine große Anzahl von Anbietern gibt, die dies völlig unabhängig tun. Im Grunde sollte jeder in der Lage sein, die Daten abzurufen. Stellen Sie sich das wie ein Garantiezertifikat vor, das bei jeder Aktion neu aufgezeichnet und veröffentlicht wird.

WAS IST WEB 3.0?

Wie funktioniert es? Kann es wirklich eine deutlich bessere digitale Zukunft für das Internet bieten?

Die Idee einer verketteten Reihe von Computern - dem Internet - ist jetzt 30 Jahre alt.

Das World Wide Web, die gemeinsame Sammlung von Websites, die wir über diese verbundenen Computer online betrachten können, wurde 1989 von Tim Berners-Lee geschaffen.

Schon in den ersten Tagen hatte er große Pläne für die Zukunft des Webs: "Das Semantische Web", das auf Seite 11 dieses Artikels erläutert wird.

In der ersten Generation des Webs, die als Web 1.0 bekannt ist, haben die Benutzer (*eigentlich Teilnehmer*) ihre eigenen Websites und E-Mails auf Servern gehostet, entweder zu Hause oder bei einer Institution. Es war eine Idylle für Hobbyisten, voller Versprechen und revolutionärem Eifer.

Am Ende konnte das nicht von Dauer sein.

Die Teilnehmer mussten sich selbst um ihre Angelegenheiten kümmern, ihre Computer ständig am Laufen halten, um ihre eigenen Webseiten und E-Mail-Postfächer zu hosten.

Das war für alle außer den engagiertesten Enthusiasten zu mühsam, so dass Ende der 1990er Jahre Unternehmen und große Organisationen einstiegen.

Dann gab es GeoCities (<https://wiki.archiveteam.org/index.php/GeoCities>), das Ihre Webseiten hostete, damit Sie es nicht tun mussten. Hotmail und Yahoo taten etwas Ähnliches für E-Mails.

Der Begriff Web 2.0 wurde 1999 geprägt, als die Menschen begannen, zu zentralisierten Anbietern zu wechseln. In dieser Ära wurden mehrere Unternehmen gegründet, die unser Leben immer stärker beeinflussen werden: YouTube (*Google-Alphabet*), Facebook - (*jetzt Meta*), Twitter (*Elon Musk*), WhatsApp (*jetzt Meta*) und Instagram (*jetzt Meta*) und mehr: Tiktok (*China*).

Der Aufstieg dieser Unternehmen hatte eine Reihe von Vorteilen: Einer davon ist, dass der Zugang zur Online-Welt deutlich einfacher geworden ist.

Wir können uns mit einem Google-Konto bei sehr vielen Websites anmelden und mit PayPal und heutzutage noch mehr Diensten online bezahlen.

Es schien alles so einfach zu werden.

Es bedeutete auch, dass es einfacher wurde den Teilnehmer oder Nutzer dieser neuen Version rigoros zu täuschen und Geld von ihm zu erpressen.

Eine Reihe neuerer Skandale zeigt, dass das Web 2.0 sehr schwerwiegende, gesellschaftsschädigende Nachteile hat.

Eines der krassesten Beispiele kam 2018, als sich herausstellte, dass das Unternehmen Cambridge Analytica die persönlichen Daten von Millionen von Facebook-Nutzern gesammelt, Profile von ihnen erstellt und ihnen dann politisch polarisierende Werbung in ihren Feeds geschickt hatte.

Das Unternehmen arbeitete für Donald Trumps Wahlkampf um die US-Präsidentschaft im Jahr 2016.





Im Jahr 2021 veröffentlichte die Whistleblowerin Frances Haugen, eine ehemalige Mitarbeiterin von Facebook, eine Reihe von Dokumenten, die ihrer Meinung nach zeigten, dass das Unternehmen seinem Wachstum Priorität einräumte und keine Garantien zum Schutz junger Menschen vor potenziell schädlichen Inhalten bot.

Im Herbst 2022 schloss Elon Musk seine Übernahme des sozialen Netzwerks Twitter ab. Innerhalb weniger Tage hatte er Mitarbeiter entlassen und die Idee lanciert, von den Nutzern Gebühren für die Verifizierung zu verlangen. Er hat in der Vergangenheit auch gesagt, dass er ein "Absolutist der freien Meinungsäußerung" ist. Für viele Nutzer, die Zeit in den Aufbau ihres Profils bei Twitter investiert hatten, war es an der Zeit zu gehen. Kurz gesagt, die rosarote Brille wurde abgenommen. Dieses Resümee zeigt, dass die Tech-Giganten zu mächtig und nicht vertrauenswürdig sind. Macht korumpiert.

Dieses Unbehagen ist einer der Gründe, warum von Web3 die Rede ist. Vielleicht haben Sie schon verstanden, was ein NFT (Non-Fungible Token) bedeutet.

(Ein nicht-fungibler Token (NFT) ist ein einzigartiger digitaler Identifikator, der nicht kopiert, ersetzt oder aufgeteilt werden kann, der in einer Blockchain aufgezeichnet wird und der dazu dient, das Eigentum und die Authentizität zu bestätigen. Das Eigentum an einer NFT wird in der Blockchain aufgezeichnet und kann vom Eigentümer übertragen werden, so dass NFTs verkauft und gehandelt werden können. NFTs können von jedermann erstellt werden und erfordern nur wenige oder gar keine Programmierkenntnisse. NFTs enthalten die Regel Verweise auf digitale Dateien wie Fotos, Videos und Audio. Da NFTs eindeutig identifizierbare Vermögenswerte sind, unterscheiden sie sich von Kryptowährungen, die fungibel sind.

Befürworter von NFTs behaupten, dass NFTs ein öffentliches Echtheitszertifikat oder einen Eigentumsnachweis bieten, aber die rechtlichen Rechte, die durch eine NFT vermittelt werden, können unsicher sein. Das Eigentum an einer NFT, wie sie von der Blockchain definiert wird, hat keine inhärente rechtliche Bedeutung und gewährt nicht notwendigerweise Urheberrechte, geistige Eigentumsrechte oder andere rechtliche Rechte an der zugehörigen digitalen Datei. Eine NFT schränkt das Teilen oder Kopieren der zugehörigen digitalen Datei nicht ein und verhindert nicht die Erstellung von NFTs, die auf identische Dateien verweisen).



<https://simon.rochester.edu/blog/deans-corner/nonfungible-tokens>

Die Universität von Rochester wird Ihnen vieles erklären: Die Verwendung von nicht fälschbaren Wertmarken (NFT) zum Verkauf von Sammlerstücken / Von Professor Derek Mohr



Illustration eines nicht-fungiblen Tokens, der von einem Smart Contract (einem Programm zur automatischen Ausführung von Vertragsbedingungen) generiert wird



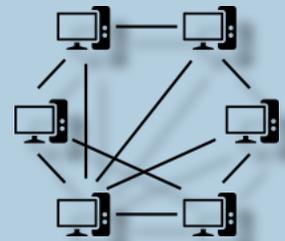


Eine Blockchain ist ein verteiltes Hauptbuch mit wachsenden Listen von Datensätzen (*Blöcken*), die über kryptografische Hashes sicher miteinander verbunden sind. Jeder Block enthält einen kryptografischen Hash des vorherigen Blocks, einen Zeitstempel und Transaktionsdaten (im Allgemeinen als Merkle-Baum dargestellt, wobei die Datenknoten durch Blätter durch Blätter dargestellt werden).



In der Kryptographie und Informatik ist ein Hash-Baum oder Merkle-Baum ein Baum, in dem in dem jedes "Blatt" (*Knoten*) mit dem kryptografischen Hash-Wert eines Datenblocks gekennzeichnet ist und jeder Knoten, der kein Blatt ist (*genannt Zweig, innerer Knoten oder Inode*), mit dem kryptografischen Hash-Wert der Kennzeichnungen seiner untergeordneten Knoten gekennzeichnet ist. Ein Hash-Baum ermöglicht eine effiziente und sichere Überprüfung des Inhalts einer großen Datenstruktur. Ein Hash-Baum ist eine Verallgemeinerung einer Hash-Liste und einer Hash-Kette.

Der Zeitstempel beweist, dass die Transaktionsdaten existierten, als der Block erstellt wurde. Da jeder Block Informationen über den vorangegangenen Block enthält, bilden sie effektiv eine Kette (vgl. Linked-List-Datenstruktur), wobei jeder weitere Block mit den vorherigen verknüpft ist. Folglich sind Blockchain-Transaktionen irreversibel, da die Daten eines Blocks, sobald sie aufgezeichnet wurden, nicht mehr rückwirkend geändert werden können, ohne alle nachfolgenden Blöcke zu verändern.



Blockchains werden in der Regel von einem Peer-to-Peer (*P2P*)-Computernetzwerk als öffentliches, verteiltes Buch verwaltet, in dem die Knoten gemeinsam ein Konsensalgorithmus-Protokoll befolgen, um neue Transaktionsblöcke hinzuzufügen und zu validieren.

Obwohl Blockchain-Datensätze nicht unveränderlich sind, da Blockchain-Forks möglich sind, können Blockchains als sicher angesehen werden und sind ein Beispiel für ein verteiltes Computersystem mit hoher byzantinischer Fehlertoleranz.

Eine Blockchain wurde von einer Person (*oder einer Gruppe von Personen*) unter dem Namen (*oder Pseudonym*) Satoshi Nakamoto im Jahr 2008 geschaffen, um als öffentliches verteiltes Hauptbuch für Bitcoin-Kryptowährungstransaktionen zu dienen, basierend auf früheren Arbeiten von Stuart Haber, W. Scott Stornetta und Dave Bayer.

Die Implementierung der Blockchain in Bitcoin machte es zur ersten digitalen Währung, die das Problem der doppelten Geldausgabe ohne eine vertrauenswürdige Autorität oder einen zentralen Server löste.

Das Bitcoin-Design hat andere Anwendungen und Blockchains inspiriert, die für die Öffentlichkeit lesbar sind und von vielen Kryptowährungen verwendet werden.

Die Blockchain kann als eine Art Zahlungsschiene betrachtet werden. Private Blockchains wurden für die geschäftliche Nutzung vorgeschlagen.

COMPUTERWORLD

(Ein digitales Magazin - <https://www.computerworld.com>) bezeichnete die Vermarktung solcher privatisierten Blockchains ohne ein angemessenes Sicherheitsmodell als "Schlangenhöl"; andere haben jedoch argumentiert, dass genehmigte Blockchains, wenn sie sorgfältig konzipiert sind, in der Praxis dezentraler und damit sicherer sein können als genehmigungsfreie.





DAS VERSPRECHEN VON WEB3

Web3 stützt sich stark auf diese "verteilten Anwendungen" - Distributed Apps -(*dapps*), um die Probleme einer zentralen Kontrolle zu vermeiden. Für Gavin Wood, einen Mitbegründer von Ethereum, der den Begriff web3 im Jahr 2014 prägte, gibt es mehrere potenzielle Vorteile.

Erstens: Da Dapps dezentralisiert sind, werden sie in gewisser Weise von jedem gehostet und nicht von einem Unternehmen - Wood hat dies als "demokratischer" bezeichnet.

Zweitens sind wir bei der Bereitstellung von Online-Diensten nicht mehr auf Unternehmen angewiesen - die Garantie kann in die zugrunde liegende Blockchain eingebaut werden.

Drittens könnte die Überprüfung der Identität von Menschen (Personen) in einem dezentralisierten Web robuster sein. Mit anderen Worten, gefälschte Konten und Bots werden sofort abgeschossen.

Die Menschen könnten immer noch anonym bleiben, aber ihre Aktivitäten könnten im Laufe der Zeit nach verfolgt werden, wiederum dank der Blockchain.

In einer idealen Zukunft könnte Web3 die Monopole der Tech-Giganten herausfordern und es viel einfacher machen, automatisierte Fake-Accounts oder Bots, die Hass-reden verbreiten, auszurotten.

Im Prinzip würde es den Nutzern auch ermöglichen, ihre Profile und Daten einfacher zwischen Online-Diensten zu verschieben, da ihre Daten auf einer gemeinsamen Blockchain und nicht auf den privaten Servern eines Unternehmens gespeichert würden.

Im Jahr 2017 haben Wood und andere die Web3 Stiftung gegründet um bei der Finanzierung und Entwicklung von Ideen zu helfen die diese neue Online-Umgebung ermöglichen könnten. Laut Alistair Stewart - leitender Forscher bei der Stiftung -

ist der grundlegende Punkt die Dezentralisierung, man will sich nicht auf einzelne Unternehmen oder sogar einzelne Personen verlassen, sondern einer breiten Gruppe von Menschen vertrauen. Es gibt einen großen Nachteil einer dezentralisierten Lösung: Die Benutzerfreundlichkeit muss im Vorfeld sehr deutlich und tiefgreifend behandelt werden, denn ein Endbenutzer verlangt immer eine einfache Bedienung, was auch für Web2 gilt.

**WENN SIE ES NICHT VERSUCHEN,
WIRD ES NICHT FUNKTIONIEREN,
BAUEN SIE ES
UND MAN WIRD KOMMEN.**

Diese technische Komplexität hat zu einem zweiten Problem geführt, das sich wie eine Wiederholung der Geschichte anfühlt: Da das Web3 davon abhängt, dass die Benutzer komplexe digitale Technologien implementieren, tauchen immer mehr Unternehmen auf, die dies für sie tun. Genauso wie Google und andere Giganten aufgetaucht sind, weil Web1 ein bisschen kompliziert war. Außerdem werden sie alles in ihrer Macht Stehende tun, um diese Version nicht erfolgreich werden zu lassen - denn das wäre ihr Ende.

Der Erfolg von Web3 hängt auch von einem Vergleich ab:

Wie genervt sind die Leute von Web2 und wie sehr wiegt das die Schwierigkeiten bei der Nutzung von Web3 auf?

Die meisten Menschen wünschen sich vor allem Benutzerfreundlichkeit, aber das lässt sich lösen. Die Benutzer mögen es zwar bedauern, dass Google oder Meta jedes Mal mehr über sie herausfinden, wenn sie sich über ihr Gmail- oder Facebook-Konto bei einer Website eines Drittanbieters anmelden, aber sie wissen die Bequemlichkeit zu schätzen und werden sich daher anpassen und klein machen müssen. Es gibt auch viele Open-Source-Optionen.

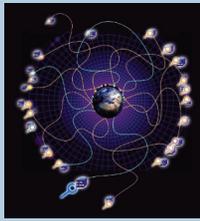
Der Markt wird zeigen, wie erfolgreich diese sind.

In jedem Fall ist es sehr wünschenswert. Wenn Sie es nicht ausprobieren, wird es nicht funktionieren, bauen Sie es und man wird kommen. Und es gibt noch eine weitere, bereits erwähnte Möglichkeit, die hinzugefügt werden muss:

das Semantic Web

(*Semantik oder Bedeutungstheorie ist eine Wissenschaft, die sich mit der Bedeutung von Symbolen befasst, insbesondere mit den Bausteinen der natürlichen Sprachen, die der Kommunikation dienen, also mit Wörtern und Sätzen*).





WEB3

Das Web 3.0 soll das Internet in eine stärker reglementierte Version verwandeln? Die Idee einer dezentralisierten Version des Internets, die den normalen Nutzern mehr Macht gibt, stößt auf großes Interesse. Darüber hinaus besteht die Möglichkeit eines Semantic Web.

Es gibt Stimmen, die aus offensichtlichen Gründen eine neue Version des Internets fordern:

- ❶ um eine deutlich bessere digitale Zukunft für das Internet zu schaffen.
- ❷ die Technologie-Giganten, die derzeit unsere Online-Welt (*Internet*) beherrschen werden von massiven Missbräuchen erschüttert, sie verlieren viel Geld - indem sie ihre Märkte schädigen - und scheinen unfähig oder nicht willens zu sein, die die Probleme der Online-Desinformation, Belästigung und Hass-reden zu lösen.

DEZENTRALISIERT?

Das bedeutet, dass es statt einiger weniger großer Anbieter, die alle (*Provider etc.*) registrierten Daten überwachen, eine große Anzahl von Anbietern gibt, die dies völlig unabhängig tun. Im Grunde sollte jeder in der Lage sein, die Daten abzurufen. Stellen Sie sich das wie ein Garantiezertifikat vor, das bei jeder Aktion neu aufgezeichnet und veröffentlicht wird.

WEB 3 stützt sich stark auf diese "verteilten Apps" (*dapps*), um die Probleme einer zentralen Kontrolle zu vermeiden.

Für Gavin Wood, einen Mitbegründer von Ethereum, der den Begriff WEB 3 im Jahr 2014 geprägt hat, gibt es mehrere potenzielle Vorteile.

Erstens,

Da Dapps dezentralisiert sind, werden sie in gewissem Sinne von jedem gehostet und nicht von einem Unternehmen - Wood hat dies als "demokratischer" bezeichnet.

Zweitens,

müssen wir uns nicht mehr auf Unternehmen verlassen, die Online-Dienste anbieten - die Garantie kann in die zugrunde liegende Blockchain eingebaut werden.

Drittens,

könnte die Überprüfung der Identität von Personen in einem dezentralisierten Web robuster sein, d.h. gefälschte Konten und Bots werden sofort abgeschossen.



<https://wehsrobotics.weebly.com/>





Semantic Web

Im Jahr 2001 schlug der Web-Gründer Tim Berners-Lee zusammen mit den Informatikern James Hendler und Ora Lassila erstmals vor, das Web so zu entwickeln, dass alle Inhalte maschinenlesbar sind - mit anderen Worten, von jedem Computer "verstanden" werden können.

Timothy John Berners-Lee ist zusammen mit seinem damaligen Manager, dem Belgier Robert Cailliau, der Schöpfer und Begründer der Technologie und der Sammlung von Vereinbarungen, die das World Wide Web oder (www) ermöglicht haben.

Er arbeitete daran, als er von Juni bis Dezember 1990 als Berater - Software-Ingenieur bei CERN in der Schweiz angestellt war. Als Direktor des World Wide Web Consortium (W3C) beaufsichtigt er die Entwicklung von Websprachen und Protokollen wie HTML, XML, CSS und HTTP.

Das bedeutet, dass das semantische Web verlangt, dass Websites Metadaten bereitstellen (→siehe Erklärung in der rechten Spalte), die dann von Software, Anwendungen und Diensten analysiert werden können.

Das Semantische Web, manchmal auch als Web 3.0 bezeichnet (nicht zu verwechseln mit Web3), ist eine Erweiterung des World Wide Web durch die vom World Wide Web Consortium (W3C) festgelegten Standards. Das Ziel des Semantic Web ist es, Internetdaten maschinenlesbar zu machen.

Jedes Element auf einer Webseite würde beschriftet werden, um das Verständnis des Computers zu erleichtern. Zum Beispiel würde der Satz "Detlef Overbeek wohnt in IJsselstein" mit sichtbaren Markierungen versehen, die dem Computer sagen, dass "Detlef Overbeek" ein Name und "IJsselstein" ein Ort ist.

Die Vorteile einer solchen Kennzeichnung aller Inhalte im Web könnten enorm sein. Da die KI immer leistungsfähiger, schneller und allgemeiner wird, würde die Verbesserung ihrer Fähigkeit, unsere Online-Wissensdatenbank zu entschlüsseln, die KI auf ein noch höheres Niveau bringen und - was entscheidend ist - die Interaktion mit der menschlichen Welt effektiver machen. In der Tat haben Apple und Google dies bereits auf Ihrem Handy eingesetzt.

Wenn Sie an einem Restaurant vorbeigehen, können sie Ihnen die Speisekarte des Restaurants, die Preise, den Chefkoch und sogar die Inneneinrichtung anzeigen. Das könnte eine sehr gute Möglichkeit sein, um sicherzustellen, dass eine Person **die Vergangenheit ist die gemeinsame Nutzung von Dokumenten, die Zukunft ist die gemeinsame Nutzung von Daten** kein Bot ist. In seiner ursprünglichen Vision des semantischen Webs stellte sich Berners-Lee

eine Situation vor, in der ein Webbrowser die Behandlungsanforderungen für einen Patienten von der Website eines Arztes abrufen und dann ein kombiniertes Angebot von z.B. Apotheken finden könnte, die die entsprechenden Medikamente anbieten, und möglicherweise Dienste, die z.B. Physiotherapie anbieten könnten.

Etwaige Termine könnten nahtlos gebucht werden, indem die Verfügbarkeit bei einem Physiotherapeuten mit dem Terminplan des Patienten verglichen wird. In Kombination mit Web3 könnte die Anmeldung bei den Patient-Arzt-Daten ein einfacher, aber gut gesicherter Schritt sein.

Metadaten sind "Daten, die Informationen über andere Daten liefern", aber nicht der Inhalt der Daten, wie etwa der Text einer Nachricht oder das Bild selbst. Es gibt viele verschiedene Arten von Metadaten, darunter:

- **Beschreibende Metadaten** - die beschreibenden Informationen über eine Ressource. Sie werden zum Auffinden und zur Identifizierung verwendet. Sie umfassen Elemente wie Titel, Kurzfassung, Autor und Schlüsselwörter.
- **Strukturelle Metadaten** - Metadaten über Datencontainer und geben an, wie zusammengesetzte Objekte zusammengesetzt sind, zum Beispiel, wie Seiten zu Kapiteln geordnet werden. Sie beschreiben die Typen, Versionen, Beziehungen und andere Merkmale von digitalen Materialien.
- **Administrative Metadaten** - die Informationen, die bei der Verwaltung einer Ressource, wie z.B. Ressourcentyp, Berechtigungen und wann und wie sie erstellt wurde.
- **Referenz-Metadaten** - die Informationen über den Inhalt und die Qualität der statistischen Daten.
- **Statistische Metadaten** - auch Prozessdaten genannt, können Prozesse beschreiben, die statistische Daten sammeln, verarbeiten oder produzieren. statistischen Daten.
- **Rechtliche Metadaten** - liefern Informationen über den Ersteller, Urheber, den Inhaber des Copyrights und die öffentliche Lizenzierung, falls vorhanden.

Metadaten sind nicht strikt an eine dieser Kategorien gebunden, da sie einen Datensatz auf viele andere Arten beschreiben können.





Die Kennzeichnung von Internetdaten auf diese Weise ist eine arbeitsintensive Aufgabe:

Nur ein winziger Bruchteil der weltweiten Webseiten enthält heute semantische Informationen.

Aber das könnte automatisiert werden. Um sie hinzuzufügen, müssen der Inhalt einer Website und der zugrunde liegende Computercode durchsucht und die Informationen dann nach ihrer Relevanz klassifiziert werden.

Diejenigen, die sich mit dieser Aufgabe beschäftigen, tun dies, weil sie das Versprechen des semantischen Webs sehen.

Jedes Jahr findet eine internationale Semantic Web-Konferenz statt; Die jüngste, die praktisch von China organisiert wurde, ging Ende Oktober zu Ende. Die Tatsache, dass es ausgerechnet China ist, das dies tut, wird uns sicher nicht gefallen. Warum nicht Europa? Warum nicht Pascal? Wir sind ständig dabei, die Sprache zu erweitern, und ich denke, das ist eine gute Idee.

Es gibt noch viel zu tun... Aber hoffen Sie deshalb auch auf das Ende des Web-Missbrauchs.

Um die Kodierung der Semantik mit den Daten zu ermöglichen, werden Technologien wie das Resource Description Framework (RDF) und die Web Ontology (-Lehre des Seins) Language (OWL) verwendet. Diese Technologien werden zur formalen Darstellung von Metadaten verwendet. Eine Ontologie kann zum Beispiel Konzepte, Beziehungen zwischen Entitäten und Kategorien von Dingen beschreiben. Diese eingebettete Semantik bietet erhebliche Vorteile, wie z.B. die Möglichkeit, Schlussfolgerungen über Daten zu ziehen und mit heterogenen Datenquellen zu arbeiten.

Diese Standards fördern gemeinsame Datenformate und Austauschprotokolle im Web, vor allem das RDF. Laut W3C "bietet das Semantic Web einen gemeinsamen Rahmen, der die gemeinsame Nutzung und Wiederverwendung von Daten über Anwendungs-, Unternehmens- und Gemeinschaftsgrenzen hinweg ermöglicht."

Das Semantische Web wird daher als Integrator für verschiedene Inhalts- und Informationsanwendungen und -systeme betrachtet.

Der Begriff wurde von Tim Berners-Lee für ein Netz von Daten (oder Daten-Netz) geprägt, das von Maschinen verarbeitet werden kann - d.h. ein Netz, in dem ein Großteil der Bedeutung maschinenlesbar ist. Während seine Kritiker seine Machbarkeit in Frage gestellt haben, argumentieren seine Befürworter, dass Anwendungen in der Bibliotheks- und Informationswissenschaft, der Industrie, der Biologie und der humanwissenschaftlichen Forschung die Gültigkeit des ursprünglichen Konzepts bereits bewiesen haben.

Berners-Lee drückte seine Vision des Semantic Web 1999 ursprünglich wie folgt aus

Ich habe einen Traum für das Web [in dem Computer] in der Lage sein werden, alle Daten im Web zu analysieren - die Inhalte, Links und Transaktionen zwischen Menschen und Computern. Ein "Semantisches Web", das dies möglich macht, muss erst noch entstehen,

aber wenn es soweit ist, werden die alltäglichen Mechanismen des Handels, der Bürokratie und unseres täglichen Lebens von Maschinen gehandhabt werden, die mit Maschinen sprechen. Die "intelligenten Agenten", von denen die Menschen seit Ewigkeiten schwärmen, werden endlich Wirklichkeit werden.

Der Scientific American Artikel von Berners-Lee, Hendler und Lassila aus dem Jahr 2001 beschrieb die erwartete Entwicklung des bestehenden Webs zu einem Semantic Web. Im Jahr 2006 stellten Berners-Lee und Kollegen fest, dass: "Diese einfache Idee...bleibt weitgehend unrealistisch". 2013 enthielten mehr als vier Millionen Webdomains (von insgesamt etwa 250 Millionen) Semantic Web Markup.





SEMANTIC WEB SOLUTIONS

Das Semantische Web beinhaltet die Veröffentlichung in Sprachen, die speziell für Daten entwickelt wurden:

Resource Description Framework (RDF), Web Ontology Language (OWL), und Extensible Markup Language (XML).

HTML beschreibt Dokumente und die Verknüpfungen zwischen ihnen. RDF, OWL und XML hingegen können beliebige Dinge wie Personen, Sitzungen oder Flugzeugteile beschreiben.

Diese Technologien werden kombiniert, um Beschreibungen bereitzustellen, die den Inhalt von Webdokumenten ergänzen oder ersetzen.

So kann sich der Inhalt als beschreibende Daten manifestieren, die in Datenbanken gespeichert sind, auf die im Web zugegriffen werden kann, oder als Markup in den Dokumenten (insbesondere in Extensible HTML (XHTML), das mit XML durchsetzt ist, oder, was häufiger vorkommt, als reines XML, wobei Layout- oder Rendering-Hinweise separat gespeichert werden).

Die maschinenlesbaren Beschreibungen ermöglichen es Content Managern, dem Inhalt eine Bedeutung zu verleihen, d.h. die Struktur des Wissens zu beschreiben, das wir über diesen Inhalt haben.

Auf diese Weise kann eine Maschine anstelle von Text das Wissen selbst verarbeiten, und zwar mit ähnlichen Prozessen wie das menschliche schlussfolgernde Denken und die Inferenz,

Dadurch erhalten sie aussagekräftigere Ergebnisse und helfen Computern bei der automatisierten Informationsbeschaffung und Recherche.

Ein Beispiel für ein Tag, das in einer nicht-semantischen Webseite verwendet werden würde:

```
<item>blog</item>
```

Die Kodierung ähnlicher Informationen in einer semantischen Webseite könnte wie folgt aussehen:

```
<item rdf:about="https://example.org/semantic-web/">Semantic Web</item>
```

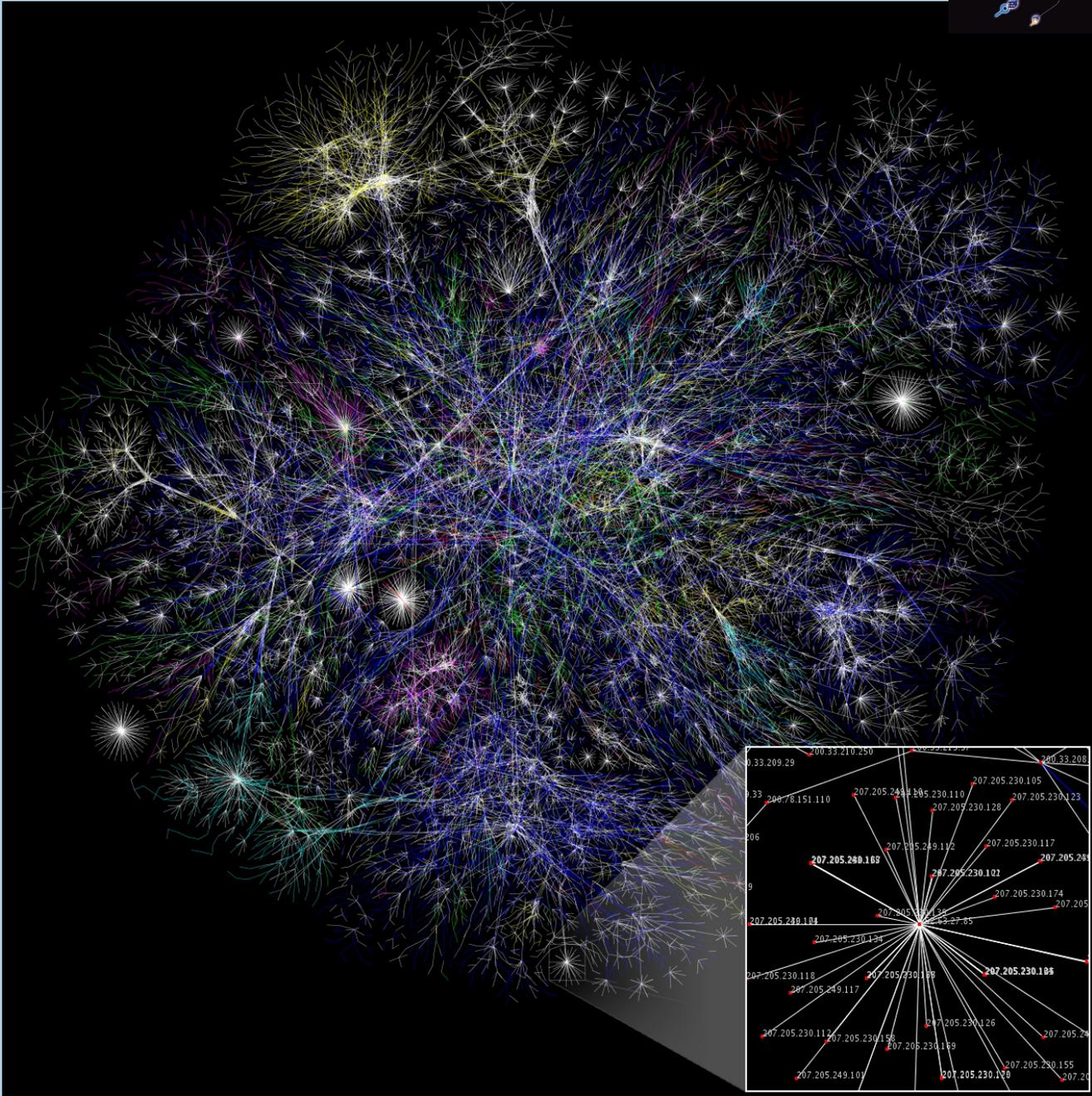
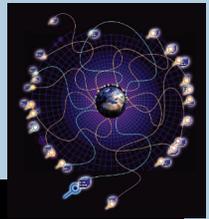
Tim Berners-Lee nennt das resultierende Netzwerk von Linked Data den **Giant Global Graph**, im Gegensatz zum HTML-basierten **World Wide Web**.

Berners-Lee vertritt die Ansicht, dass, wenn die Vergangenheit das Teilen von Dokumenten war, die Zukunft das Teilen von Daten ist.

Seine Antwort auf die Frage nach dem "Wie" enthält drei Hinweise.

- ❶ eine URL sollte auf die Daten verweisen.
- ❷ jeder, der auf die URL zugreift, sollte die Daten zurückerhalten.
- ❸ Beziehungen in den Daten sollten auf weitere URLs mit Daten verweisen.





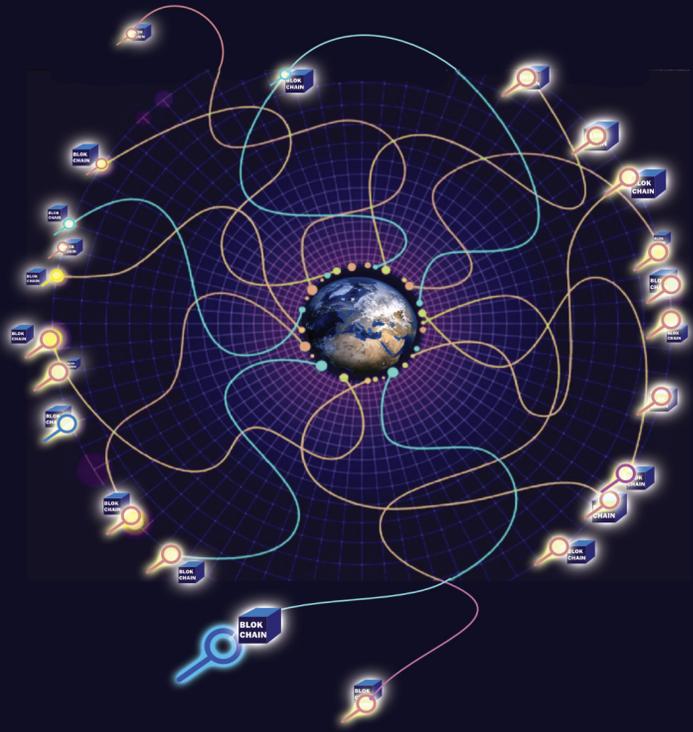
Copyright: <https://commons.wikimedia.org/w/index.php?curid=1538544>



- Page 1
- Page 2
- Page 3
- Page 4
- Page 5
- Page 6
- Page 7
- Page 8
- Page 9
- Page 10
- Page 11
- Page 12
- Page 13
- Page 14
- Page 15
- Page 16
- Page 17
- Page 18

BLAISE PASCAL MAGAZINE 109

Multi platform /Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



BLAISE PASCAL MAGAZINE

procedure
begin
end

Prof. Dr. Werth, Creator of Pascal Programming language

Blaise Pascal, Mathematician

BLAISE PASCAL MAGAZINE

procedure
begin
end

Prof. Dr. Werth, Creator of Pascal Programming language

Blaise Pascal, Mathematician

Editor in Chief: Dettlef Overbeek
Edelstenenbaan 21 3402 XA
IJsselstein Netherlands

Prof. Dr. Werth, Creator of Pascal Programming language

editor@blaisepascalmagazine.eu
https://www.blaisepascalmagazine.eu

~~€ 100~~

Special offer € 75

More than 6500 pages
Issue Nr 1 bis 109 all
English spoken

<https://www.blaisepascalmagazine.eu/product/lib-stick/>

SEARCH PDF: BLAISEPASCALMAGAZINE_109_UK.PDF

Load PDF...



140%

indexer

Search in PDF

Click a result to view the corresponding result.

Page 47:
d German • Free Lazarus PDF Kit
indexer • 8
Issues per year • minimal 6

Page 48:
we showed how to create a PDF
indexer : a program to analyse a bunch

Page 51:
Search class is part of the fp
indexer unit presented in the previous

Page 54:
d in the indexed PDFs. The PDF
indexer stores for each word the name

Page 66:
INCLUDING THE NEW LAZARUS
KIT PDF **indexer**

BLAISE PASCAL MAGAZINE 109
Multi platform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js / Databases / CSS Styles / Progressive Web Apps / Android / iOS / Mac / Windows & Linux

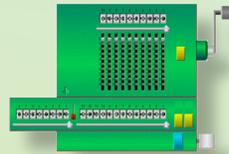
THE BLAISE PASCAL LIBRARY (THE LIBSTICK LIBRARY ON USB STICK) INCLUDES NOW THE PDF INDEXER



More than 6500 pages Issue Nr 1 to 109

<https://www.blaisepascalmagazine.eu/product/lib-stick/>

EIN GANZZAHLIGER (INTEGER) QUADRATWURZEL-ALGORITHMUS



7 Starter Expert  

EINFÜHRUNG

Die Programmiersprache Delphi verfügt über die Operatoren `div` und `mod` für ganzzahlige Divisions- und Restwertberechnungen. Für die Quadratwurzel gibt es jedoch nur Fließkommaberechnungen. Dieses Projekt stellt einen ganzzahligen Quadratwurzel-Algorithmus vor, der auch den Rest liefert.

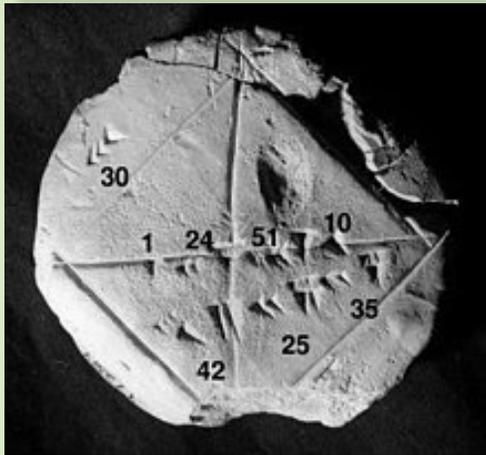


Figure 1: The Yale Babylonian Collection YB C 7289 clay tablet was created between 1800 BC and 1600 BC.

Wir schreiben also für die Quadratwurzel der Zahl N :
 $N = \text{root}^2 + \text{remainder}$.

Bitte sehen Sie sich die Bilder unten an:

Was ist die Quadratwurzel? Die Quadratwurzel einer Zahl ist definiert als der Wert, den die Zahl ergibt, wenn sie mit sich selbst multipliziert wird. Das Radikalsymbol $\sqrt{\quad}$ wird verwendet, um die Quadratwurzel anzugeben. Zum Beispiel: $\sqrt{16} = 4$.

`div` und `mod`

Division (Mathematik), die mathematische Operation, die die Umkehrung der Multiplikation ist.

Mod (modulus) dividiert zwei Zahlen und gibt nur den Rest zurück. Zum Beispiel würde der Ausdruck "a:= 13 mod 4;" 1 ergeben (a=1), während "b := 12 mod 4;" 0 ergeben würde (b=0).

" Das Vorzeichen des Ergebnisses eines Mod-Operators ist das gleiche wie das Vorzeichen des linken Operanden des Mod-Operators. Tatsächlich ist der Mod-Operator äquivalent zu der folgenden Operation:"

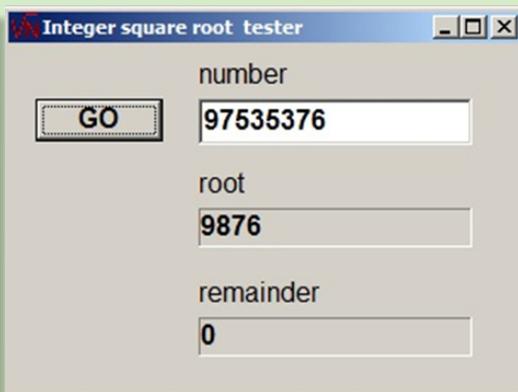
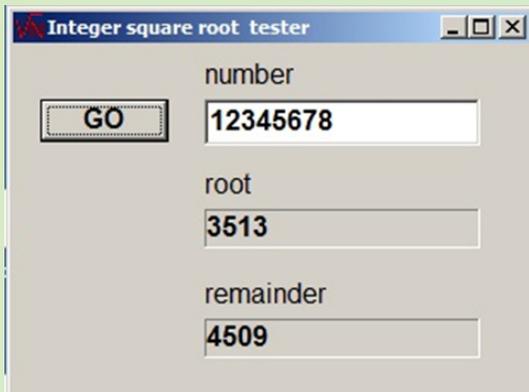
Das reservierte Wort `mod` gehört zu den Operatoren.

Das reservierte Wort `mod` wird verwendet, um ganze Zahlen zu dividieren.

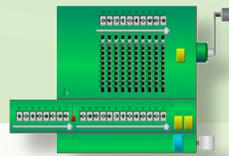
Das reservierte Wort `mod` steht für modulo und gibt den Rest einer Division zurück.

Beispiel:

```
var
  intI: integer;
begin
  ...
  intI := 10 mod 3; // Result: 1
  ...
end;
```



EIN GANZZAHLIGER (INTEGER) QUADRATWURZEL-ALGORITHMUS



DER ALGORITHMUS

Das Integer-Quadratwurzel-Projekt verwendet positive 32-Bit-Ganzzahlen (*kardinal, dword*) für N und positive 16-Bit-Ganzzahlen (*word*) als Wurzel und Rest.

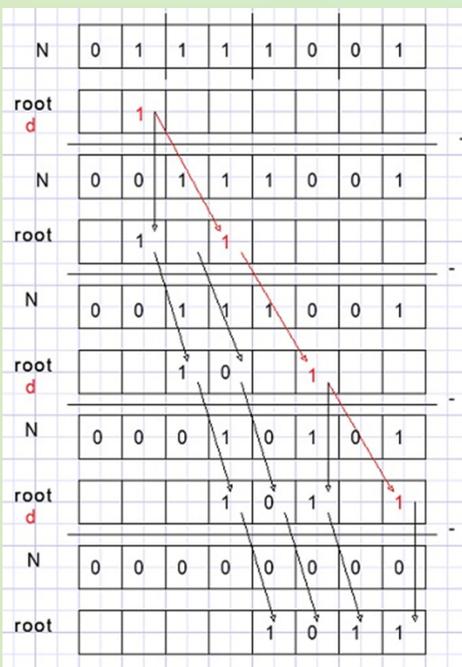
Zur Verdeutlichung wird ein Beispiel gegeben, bei dem N ein Byte ist.
Wir berechnen die Quadratwurzel aus $121 = 5^2 = [0111\ 1001]_2$.

Zunächst wird die Binärzahl N in Gruppen von 2 Bits organisiert, die von der rechten Seite aus gezählt werden. Die Variablen N , $root$ und d sind allesamt Bytes.

d ist ein Einzelbit wert (*rot dargestellt*) und befindet sich anfangs rechts in der äußersten linken 2-Bit-Gruppe von N .

Die Variable $root$ ist anfangs Null (schwarz dargestellt).
In der Abbildung unten sind leere Zellen gleich Null.

Der Wert $root + d$ wird nur dann von N subtrahiert, wenn $N \geq root+d$ ist.



Nach einer (*versuchten*) Subtraktion:

- wenn eine Subtraktion stattgefunden hat, wird das d -Bit in die Wurzel kopiert
 - wird die Wurzel um ein Bit nach rechts verschoben
 - wird das d -Bit um 2 Bits nach rechts verschoben
- Dieser Prozess wird fortgesetzt, bis das d -Bit aus dem Byte heraus geschoben ist.
Dann enthält N den Rest.

WIE DIES FUNKTIONIERT

Betrachten Sie eine Binärzahl mit 2 Bits a, b : $[ab]_2 = 2a+b$.

Das Quadrat ist $(2a+b)^2 = 4a^2 + 4ab + b^2$

Dieses Quadrat ist also eine Vier-Bit-Zahl.

Jetzt, $a^2 = a$, $b^2 = b$ für einzelne Bits.

Für $a = 1$: $4a^2 = [100]_2$ und $4ab+b^2 = [101]_2$ if $b=1$.

If $N \geq [100]$ then $[100]$ ist subtrahiert von N und $a=1$

Bleibt noch, den Wert von b zu finden.

Für $a=1$ muss $4ab = [101]$ von N subtrahiert werden, wenn $b=1$ ist.

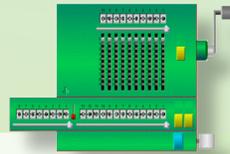
Dies gilt allerdings nur für 4-Bit-Zahlen:

Diese Bits können als die 4 oberen Bits einer größeren Zahl betrachtet werden.

Nach jedem (versuchsweisen) Subtraktionsschritt, bei dem Bit b gefunden wird, wird a ersetzt durch $[ab]_2$.

Dann a^2 wurde bereits abgezogen von N also ein nächstes bit von b muss durch versuchsweise Subtraktion von $[a01]_2$.





Das Programm

N ist die (32 Bit) Zahl, aus der die Wurzel berechnet wird.

Am Ende enthält N den Rest.

Die Variable root enthält den Wert von a.

Die Variable d enthält den Wert von b.

Zu Beginn wird a auf Null gesetzt, d wird für die erste Subtraktion gesetzt, um a zu finden.

Dann enthält root den Wert von a.

Nachfolgend sehen Sie die Prozedur zur Berechnung der ganzzahligen Wurzel:

```
const chrset = ['0'..'9',#08,'$'];

function IntRoot(N : dword; var rem : word) : word;
  //return integer square root of N and remainder rem
var d,root,sub : dword;
begin
  d := $40000000;
  root := 0;
  repeat
    sub := root or d;

    if sub <= N then
      begin
        N := N - sub;
        root := (root shr 1) or d;
      end
    else root := root shr 1;

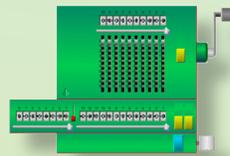
  d := d shr 2;
  until d = 0;
  result := root;
  rem := N;
end;

procedure TForm1.GoBtnClick(Sender: TObject);
var N,root : dword;
    rem : word;
begin
  if length(numberEdit.Text) = 0
  then numberEdit.Text := '0';

  N := strtoint(numberEdit.Text);
  root := IntRoot(N,rem);
  rootLabel.Caption := inttostr(root);
  remainderlabel.Caption := inttostr(rem);
end;

procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key in chrset then
    begin
      if (key = '$') and (length(numberEdit.Text) > 0)
      then key := #0;
    end
  else key := #0;
end;
```





7 Starter Expert

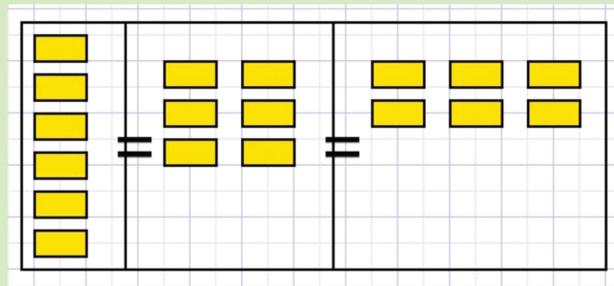
Factorising a number means writing the number as the product of its prime factors.

Example: $3960 = 2^3 \cdot 3^2 \cdot 5 \cdot 11$

If a quantity can be factorised, it means that it may be rearranged

(from a single column into 2 columns and even three.)

See picture below and the explanation at right



EXPLANATION OF FACTORISING

Can we divide 3960 exactly by 2?

$$3960 \div 2 = 1980$$

Yes it can.

The answer should be a whole number, and 1980 is $990 : 2 (2^3)$

Let's try the next prime number, 3:

$$495 : 3 = 165$$

$$165 : 3 = 55 (3^2)$$

until 10 it doesn't work

But 11 does, so we get: 5

so here comes:

$$3960 = 2^3 * 3^2 * 5 * 11$$

$$2^3 = (2 * 2 * 2) = 8 \quad 3^2 = (3 * 3) = 9$$

$$8 * 9 * 5 * 11$$

Such a rearrangement is not possible for a prime number.

Exchanging banknotes is an application of factorization in real life.

Simplifying a fraction by dividing both numerator and denominator by common factors is another example.

The common factors of integers

A and B is written as $GCD(A, B)$ {Greatest Common Divisor}

$LCM(A, B)$ yields the smallest number

that is a multiple of both A and B {least Common Multiple}

$$GCD(A, B) * LCM(A, B) = A * B$$

Problem 1

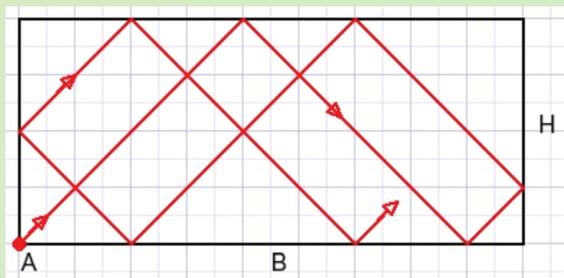
A supermarket has 255 oranges and 595 apples from which they want to make as much as possible equal packages. Since $GCD(255, 595) = 85$ $85(3 + 7)$ is the answer.

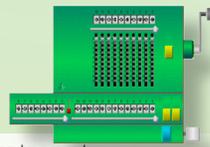
(85 packages of each 3 oranges and 7 apples)

Problem2

A light beam travels in a room with mirrors.

Which distance is travelled when the beam hits home? (At point A).





Beobachten Sie die horizontale und vertikale Bewegung getrennt.
 Nach einer vertikalen Bewegung von $2H$ befindet sich der Balken am unteren Rand.
 Nach einer horizontalen Bewegung von $2B$ befindet sich der Balken auf der linken Seite des Rechtecks.
 Die Rückkehr zum Punkt A erfolgt nach einer gemeinsamen Bewegung von $\text{LCM}(2B, 2H)$ Distanzen.
 Für das obige Bild mit $B=9$; $H=4$
 $\text{LCM}(18,8) = 72$, was der Entfernung 36 sowohl horizontal als auch vertikal entspricht.
 Die tatsächliche Entfernung ist (die Quadratwurzel aus 2) mal 36.

DAS FAKTORISIERUNGSPROGRAMM

Die ersten Primfaktoren sind 2 und 3.

Größere Primfaktoren sind von der Form $6K-1$ und $6K+1$, wobei $K=1, 2, 3, \dots$

denn $6K, 6K+2, 6K+3, 6K+4$ können keine Primfaktoren sein. Die Faktorisierung einer Zahl (N) erfordert die Division durch alle möglichen Primfaktoren und das Zählen der Divisionen, die einen Rest von Null haben. Nach dem Ausprobieren der Faktoren 2,3,5 werden die Faktoren abwechselnd durch 2 und 4 aktualisiert (von $6K-1$ über $6K+1$ bis $6K+5$).

Das spart Zeit. Eine weitere zeitsparende Überlegung ist die folgende:

$N = Q \cdot d + r$ {N: Zahl; d: Divisor; Q: Quotient; r: Rest}

Wenn $Q \leq d$ ist, werden bei weiteren Divisionen keine neuen Faktoren gefunden.

Wenn f ein Faktor von N ist, so wird $N = Q \cdot f$ für $f < Q$ gefunden.

In der Prozedur factorize wird das XF (exit flag) auf true gesetzt, um den Faktorisierungsprozess zu beenden.

```

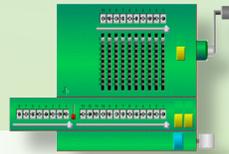
procedure factorize(N : dword);
var N1 : dword;
    count : byte; // same factors
    factor,m : word;
    fplus : byte;
    XF : boolean;
begin
    factor := 2;
    fplus := 2;
    repeat
    count := 0;
    repeat
    m := N mod factor;
    N1 := N div factor;
    XF := N1 <= factor;
    if m = 0 then
    begin
    N := N1;
    inc(count);
    end;
    until m > 0;

    if count > 0 then report(factor,count);

    if factor < 5
    then factor := (factor shl 1) - 1
    else
    begin
    factor := factor + fplus;
    fplus := 6 - fplus;
    end;
    until XF;
    if N > 1 then report(N,1);
end;
  
```

Die Report-Prozedur fügt den Faktor und die Anzahl zu einer TMemo-Komponente hinzu.
 Der GCD von 2 Zahlen lässt sich aus ihrer Faktorisierung ermitteln.
 Multiplizieren Sie einfach die gemeinsamen Faktoren.
 Es gibt jedoch eine bequemere Methode: die Anweisung (Theorem) von Euklid.





EUKLIDES

Das euklidische Theorem lautet $GCD(A,B) = GCD(A-B,B) \{A > B\}$
 Die obige Operation erzeugt also keine neuen Faktoren und es geht auch kein Faktor verloren.
 Die nächste Prozedur berechnet den GCD gemäß diesem Theorem:

```
function GCD(A,B : dword) : dword;
var H : dword;
begin
    while A > 0 do
        begin
            if A < B then
                begin
                    H := A;
                    A := B;
                    B := H;
                end;
            A := A mod B;
        end;
    result := B;
end;
```

Weitere Informationen finden Sie im Quellcode.

EUKLIDISCHE GEOMETRIE

Die euklidische Geometrie ist ein mathematisches System, das dem alexandrinischen griechischen Mathematiker Euklid zugeschrieben wird und das er in seinem Lehrbuch der Geometrie, den Elementen, beschrieben hat (wenn auch nicht streng nach modernen Standards). Euklids Methode besteht darin, mit einem kleinen Satz intuitiv ansprechender Axiome zu beginnen und daraus viele weitere Sätze (Theoreme) abzuleiten.

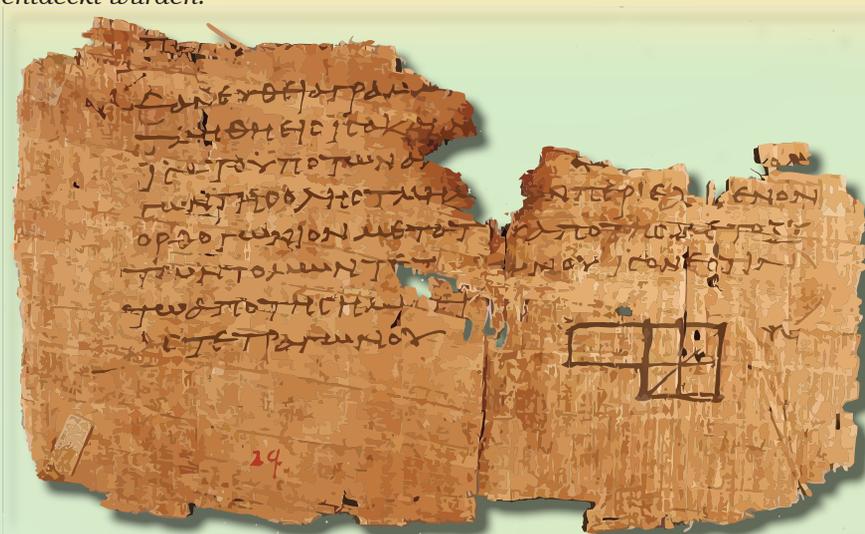
Obwohl viele von Euklids Ergebnissen bereits von früheren Mathematikern aufgestellt worden waren,

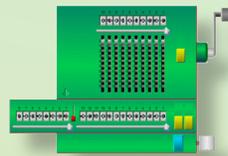
Euklid war der erste, der zeigte, wie sich diese Sätze in ein umfassendes deduktives und logisches System einfügen ließen. Die Elemente beginnen mit der ebenen Geometrie, die immer noch in den weiterführenden Schulen als das erste axiomatische System und die ersten Beispiele für formale Beweise unterrichtet wird.

Es geht weiter mit der festen Geometrie in drei Dimensionen. Ein großer Teil der Elemente enthält Ergebnisse aus dem, was man heute Algebra und Zahlentheorie nennt, die in geometrischer Sprache erklärt werden. Über zweitausend Jahre lang war das Adjektiv "euklidisch" überflüssig, weil keine andere Art von Geometrie erdacht worden war. Euklids Axiome schienen so intuitiv offensichtlich zu sein (mit der möglichen Ausnahme des Parallelitätspostulats), dass jedes aus ihnen bewiesene Theorem in einem absoluten, oft metaphysischen Sinne als wahr angesehen wurde. Heute sind jedoch viele andere nicht-euklidische Geometrien bekannt, von denen die ersten im frühen 19. Jahrhundert entdeckt wurden.



WIKIPEDIA

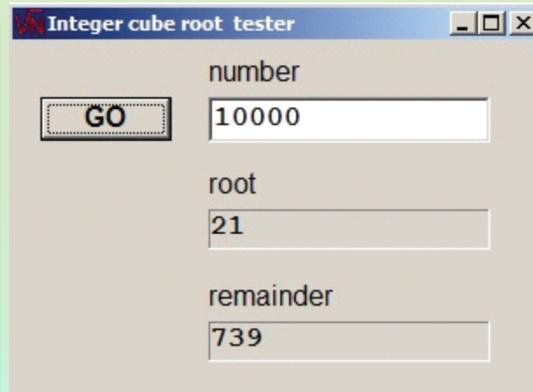
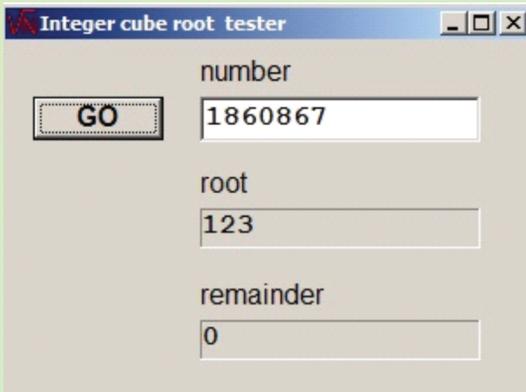




Dieser Artikel weist Ähnlichkeiten mit dem zuvor beschriebenen
Quadratwurzel-Algorithmus auf.

Zahl $N = \text{Wurzel}^3 + \text{Rest}$.

Die Funktion `cuberoot` gibt die (Kubik-)Wurzel und den Rest von N zurück.
 N wird als Gruppe von 3 Bits betrachtet, gezählt von der rechten Seite.



THEORY

$$[ab]_2 = 2a + b$$

$$(2a + b)^3 = 8a^3 + 12a^2b + 6ab^2 + b^3$$

Zu Beginn ist $a = 0$ und 1 wird vom ganz rechten Bit der höchsten Gruppe von N subtrahiert.
Bei Erfolg ist $a = 1$.

Der Begriff $8a^3 = [1000]_2$ wurde subtrahiert von N .

Das nächste Bit b muss gefunden werden, also $12ab^2 + 6ab^2 + b^3$
muss subtrahiert werden von N .

Wenn eine Subtraktion möglich wäre, wird ein $[a1]_2$ sonst wird a zu $[a0]_2$

Um das nächste b -Bit bei einem bestimmten Wert von a zu finden, muss man also die
Subtraktion von $12a^2b + 6ab^2 + b^3$ und weil $b = 1$ getestet wird, wird der Subtrahend:

$$12a^2 + 6a + 1 = [a0]_2 ([a0]_2 + 1) * 3 + 1$$

Hinweis: $[a0]_2 = 2a$

Jeder Subtraktionsversuch liefert ein weiteres Bit für die Wurzel.

Bei 30-Bit-Zahlen werden diese Schritte 10 Mal wiederholt, um ein 10-Bit-Ergebnis zu
erhalten.

Hinweis: Kubikwurzeln können auch aus negativen Zahlen gezogen werden.



THE PROGRAM

```

function CubeRoot(N : longInt; var rem : longInt) : smallInt;
// -1,073,741,823 <= N <= 1,073,741,823
// return integer cube root of N and remainder rem
const max = 1073741824;

var NX,sub,r2 : longInt;
    root : word;
    p : shortInt;
    neg : boolean;

begin
  if (N <= -max) or (N >= max) then
    begin
      result := 0;
      rem := N;
      exit;
    end;

  if N < 0 then
    begin
      N := -N;
      neg := true;
    end
  else neg := false;

  p := 30;
  root := 0;

  repeat
    r2 := root shl 1;
    sub := r2*(r2+1)*3+1;
    sub := sub shl p;
    root := root shl 1;
    NX := N - sub;

    if NX >= 0 then
      begin
        N := NX;
        root := root or 1;
      end;

    p := p - 3;
    until p < 0;

    if neg then
      begin
        root := -root;
        N := -N;
      end;

    result := root;
    rem := N;
  end;

procedure TForm1.GoBtnClick(Sender: TObject);

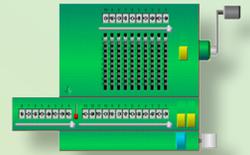
var N, rem : longInt; root : smallInt;

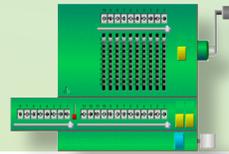
begin
  if length(numberEdit.Text) = 0 then
    numberEdit.Text := '0';

  N := strtoint(numberEdit.Text);
  root := cubeRoot(N,rem);
  rootLabel.Caption := inttostr(root);
  remainderlabel.Caption := inttostr(rem);

end;

```





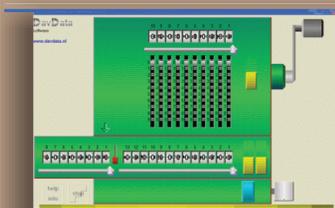
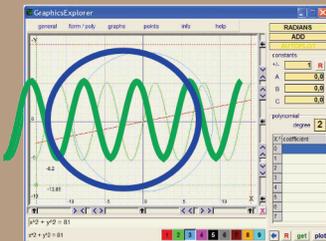
```
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
var OK,mt : boolean;

begin
  OK := false;
  with numberEdit do
    begin
      mt := length(text) = 0;
      case key of
        #8 : OK := true;
        '1'..'9' : OK := true;
        '0' : OK := not(mt);
        '$' : OK := mt;
        'a'..'f' : OK := text[1] = '$';
        '.' : OK := mt;
      end;
      if OK = false then key := #0;
    end;
  end;
end;
```



DAVID DIRKSE

including 50 example projects



COMPUTER (GRAPHICS) MATH & GAMES IN PASCAL

<https://www.blaisepascalmagazine.eu/product-category/books/>



LAZARUS HANDBOOK

2 POCKETS

Price: € 26,50

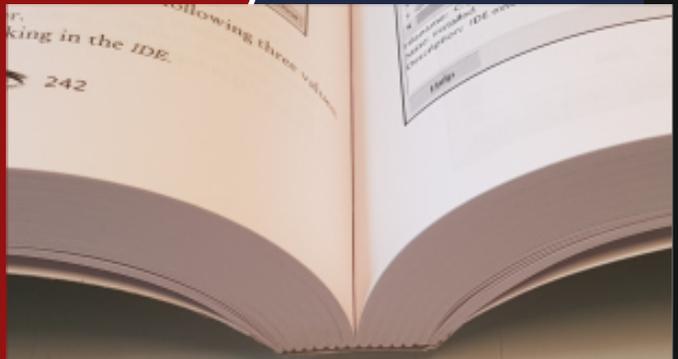
Excluding VAT and Shipping

934 PAGES

PDF & INDEX

INCLUDED

- English
- Printed black & white
- 2 Volumes
- PDF included
- 934 Pages
- Weight: 2kg
- Extra protected
- Including 40 Example projects and extra programs



BlaisePascalMagazine
PDF viewer included

DER LAZARUS DEBUGGER

- Eine Einführung und Anleitung von Martin Friebe

The Lazarus debugger: Brings your application back to life.



SEITE 1/7



TEIL 1: EINLEITUNG UND ERSTE SCHRITTE WAS IST EIN DEBUGGER?

Viele Programmierer verwenden - wenn sie ihren ersten Code schreiben und auf ihren ersten Fehler stoßen - das "writeln"-Debugging, um den fehlerhaften Code zu finden. Das bedeutet, dass Sie alle paar Zeilen "writeln"-Anweisungen in den Code einfügen, ihn neu kompilieren und dann ausführen. Schauen Sie sich die Ausgabe an, verfeinern Sie sie und wiederholen Sie den Vorgang. Bis schließlich der falsche Code gefunden wird. Danach ist Aufräumen angesagt und alle writeln müssen wieder entfernt werden.

Ein Debugger kann bei genau denselben Schritten helfen, erfordert aber nicht, dass der Code geändert wird. Und ein Debugger kann noch viel mehr als nur die oben genannten Schritte ausführen. In diesem Lernprogramm werden wir uns die Verwendung eines Debuggers ansehen. Der erste Artikel zeigt, wie Sie writeln ersetzen können. In weiteren Artikeln werden wir dann entdecken, welche anderen Werkzeuge der **Lazarus Debugger** bietet.

EINRICHTEN IHRER IDE UND IHRES PROJEKTS FÜR DIE FEHLERSUCHE

Wir beginnen mit einem kurzen Blick auf die minimal erforderlichen Einstellungen.

Wir werden uns in einem späteren Teil der Serie mit weiteren Details befassen.

Die IDE wird mit Standardeinstellungen für das Debugging ausgeliefert, die Sie so belassen können, wie sie sind. Wenn Sie jedoch ein Upgrade von älteren Lazarus-Versionen durchgeführt haben oder selbst Änderungen an diesen Einstellungen vorgenommen haben, sollten Sie sie überprüfen.

Dazu müssen Sie in der IDE die Option Optionen aus dem Menü Werkzeug wählen.

Die entsprechenden Einstellungen finden Sie auf der Seite Debugger → Debugger Backend. Siehe Abbildung 1 auf dieser Seite. Wenn Sie unter Windows oder Linux arbeiten, sollte die Auswahl oben auf der Seite "FpDebug [FpDebug internal Dwarf-debugger]" lauten. Der wichtige Teil ist der Name im Quadrat brackets. Wenn Sie einen Mac verwenden, sollte es "[LLDB Debugger (mit fpdebug) (Beta)]" heißen.

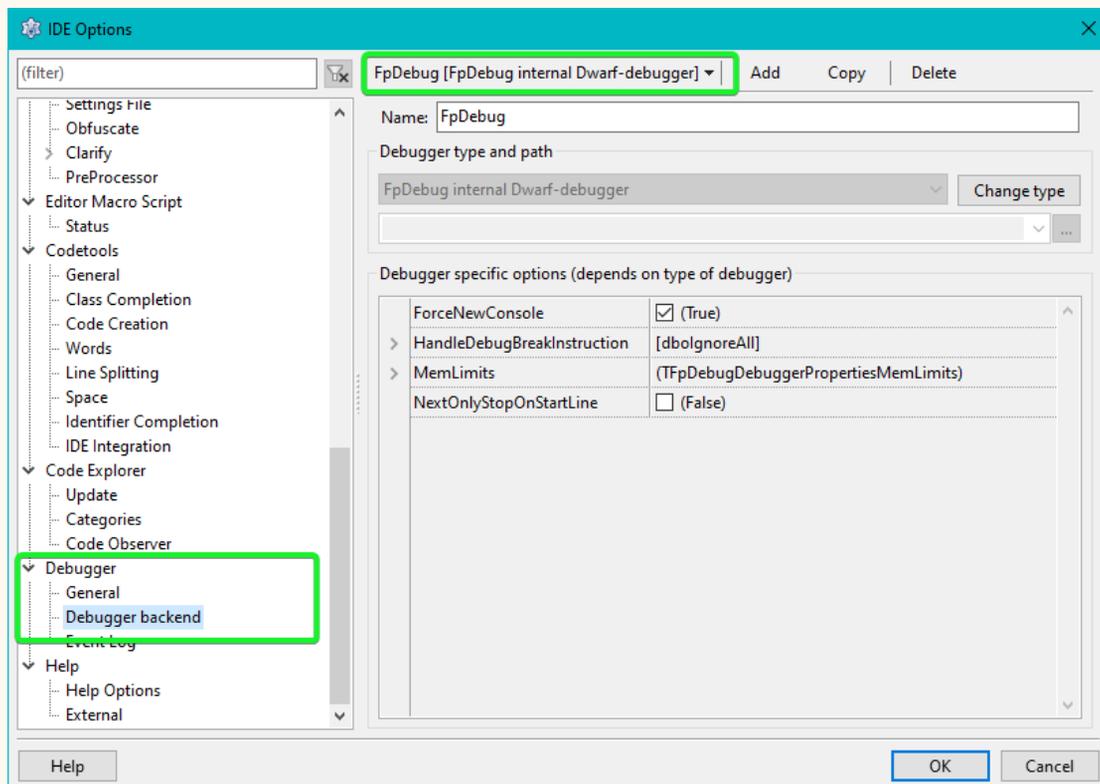


Abbildung 1





Es gibt auch Einstellungen für jedes Projekt, die festgelegt werden müssen. Auch hier sind die meisten von ihnen bereits standardmäßig eingestellt. Siehe Abbildung 2 auf dieser Seite, die Einstellung "Info für Debugger generieren" muss aktiviert werden. Auch die "Debugger-Info", die auf "Standard" eingestellt ist, sollte auf "Dwarf 3" geändert werden.

Solange die Generierung jedoch aktiviert ist, wird die IDE Sie später auffordern, einen Wert auszuwählen, wenn Sie diese Änderung nicht vornehmen (Abbildung 3).

Wenn dieser Dialog erscheint, wählen Sie "Zwerg 3".

Beachten Sie, dass das Layout des Dialogs in Lazarus 2.4 anders aussehen wird.

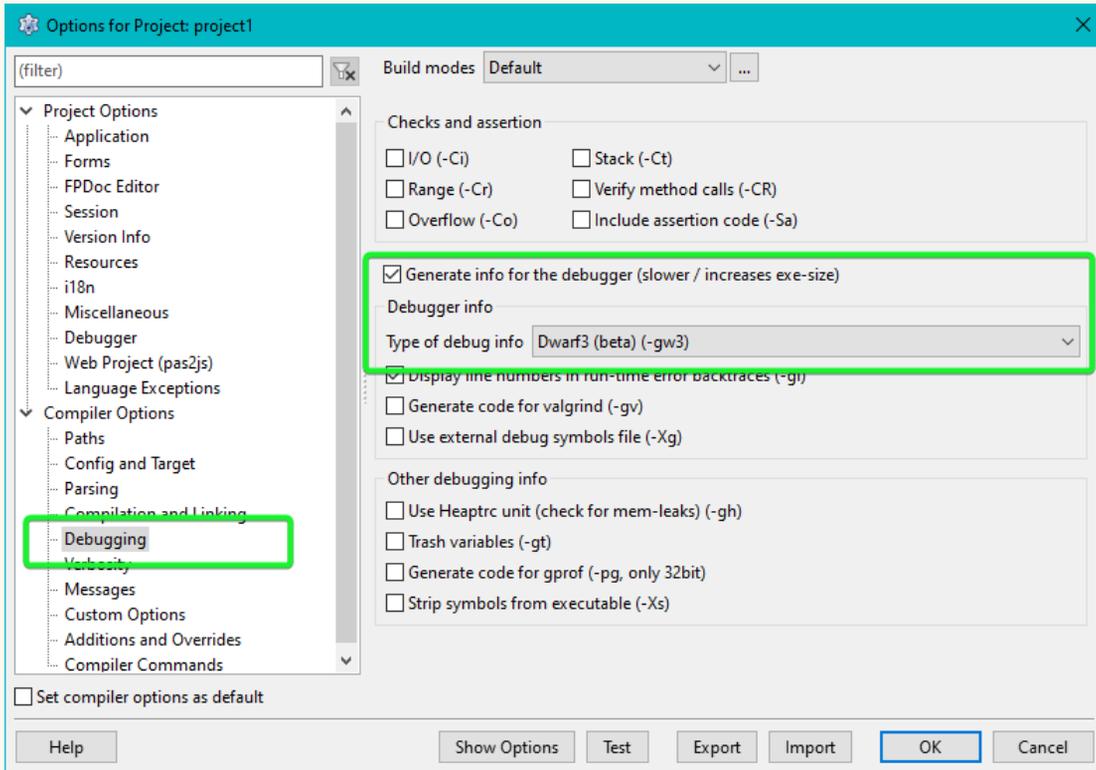


Abbildung 2

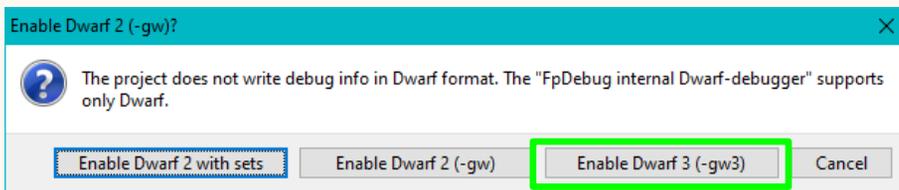


Abbildung 3





Beginnen wir mit dem Debugging

Für diese Sitzung werden wir das folgende Beispiel debuggen

```

1. program example_1;
2. const SAMPLE_TEXT = 'Just some random text and some letters A B C';
3.
4. function FindNextWord(AInput: Ansistring; ACurrentWordStart: Integer;
5.   out ANextStart, ANextLength: Integer): Boolean;
6. var
7.   AInputLength: SizeInt;
8. begin
9.   ANextStart := ACurrentWordStart;
10.  while (ANextStart > 0) and
11.    (ANextStart < Length(AInput)) and
12.    (AInput[ANextStart + 1] <> ' ')
13.  do
14.    inc(ANextStart);
15.  inc(ANextStart);
16.  while (ANextStart < Length(AInput)) and
17.    (AInput[ANextStart] = ' ')
18.  do
19.    inc(ANextStart);
20.
21.  Result := ANextStart <= Length(AInput);
22.  if not Result then
23.    exit;
24.
25.  ANextLength := 0;
26.  AInputLength := Length(AInput);
27.  while (ANextStart + ANextLength < AInputLength) and
28.    (AInput[ANextStart + ANextLength] <> ' ')
29.  do
30.    inc(ANextLength);
31. end;
32.
33. var
34.  AStart, ALen: Integer;
35.  AWord: Ansistring;
36. begin
37.  while FindNextWord(SAMPLE_TEXT, AStart, AStart, ALen) do begin
38.    AWord := copy(SAMPLE_TEXT, AStart, ALen);
39.    writeln(AWord);
40.  end;
41.  ReadLn;
42. end.

```

Es zerlegt den Text in Wörter oder Buchstaben, die durch Leerzeichen getrennt sind. Dann wird jedes Wort in die Konsole geschrieben. Und hier ist die Ausgabe:

```

Just
some
random
text
and
some
letters

```

```

A
B

```

← Empty Last Line





Sieht gut aus... Außer, dass die letzte Zeile falsch ist. Dort sollte ein "C" stehen, aber es ist leer.

Um das Problem zu beheben, überlegen wir uns, wo wir das writeln einfügen würden, d.h. wo wir den Status - z.B. den Wert einer Variablen - des Programms sehen wollen. Wir möchten vielleicht die Werte von AStart und ALen nach jedem Aufruf von FindNextWord beobachten. Damit der Debugger uns diese Werte anzeigt, müssen wir die Anwendung anhalten, und das können wir dem Debugger mitteilen, indem wir einen Haltepunkt setzen. Dazu klicken wir mit der linken Maustaste auf den grauen Bereich links vor den Zeilennummern. Ein roter Punkt wie in Abbildung 4 auf dieser Seite wird angezeigt.

```

.   while FindNextWord(SAMPLE_TEXT, AStart, AStart, ALen) do begin
.   AWord := copy(SAMPLE_TEXT, AStart, ALen);
39  writeln(AWord);
40  end;

```

Abbildung 4

Jetzt können wir die Anwendung mit F9 oder der grünen Pfeiltaste auf dem  führen. Der rote Punkt - der den Haltepunkt anzeigt - ändert sich, wenn die Programmausführung diese Zeile erreicht.

Ein grüner Pfeil im Haltepunkt (Abbildung 5 auf dieser Seite) und eine graue Hintergrundfarbe für die Zeile zeigen uns, dass der Debugger die Anwendung in dieser Zeile angehalten hat.

```

.   begin
.   while FindNextWord(SAMPLE_TEXT, AStart, AStart, ALen) do begin
.   AWord := copy(SAMPLE_TEXT, AStart, ALen);
39  writeln(AWord);
40  end;

```

Abbildung 5

Jetzt wollen wir sehen, wie die Werte von AStart und ALen lauten. Dies können Sie tun, indem Sie den Mauszeiger über eine der Variablen halten und warten, bis der Hinweis erscheint (Abbildung 5 auf dieser Seite). Der Hinweis zeigt den aktuellen Wert der Variable an.

Da wir FindNextWord zum ersten Mal ausgeführt haben, erwarten wir das erste Wort, das an Position 1 in der Zeichenkette beginnt. Der Wert, den wir sehen, ist daher korrekt.

```

AStart, ALen);
AStart = 1
var AStart: Integer = longint
project1.lpr(34,3)

```

Abbildung 6

Wir können auch das Fenster Überwachen öffnen. Strg-Alt-W oder über das Menü Ansicht > Debugfenster > Überwachen. Über den Button + können wir die 2 Variablen mit ihren Namen hinzufügen: AStart und ALen (Abbildung 7 auf dieser Seite). Wir sehen den gleichen Wert für AStart. Und für ALen sehen wir 4, was mit dem ersten Wort "Just" korreliert.

Expression	Value
AStart	1
ALen	4
AWord	'Just'

Abbildung 7





Da die Werte den Erwartungen entsprechen, können wir **Run/F9** drücken und das Programm weiterlaufen lassen. Es wird erneut angehalten, wenn der Haltepunkt bei der nächsten Iteration der Schleife erreicht wird. Dieses Mal sind die Werte `AStart=6` und für `ALen=4` für das Wort "some". Auch das ist korrekt. Wir setzen dies für jedes Wort fort und überprüfen die Werte. Schließlich sollten wir den Buchstaben "C" erwarten, der in der Ausgabe fehlt. Dazu zeigt uns das Uhrenfenster (siehe *Abbildung 8 auf dieser Seite*):

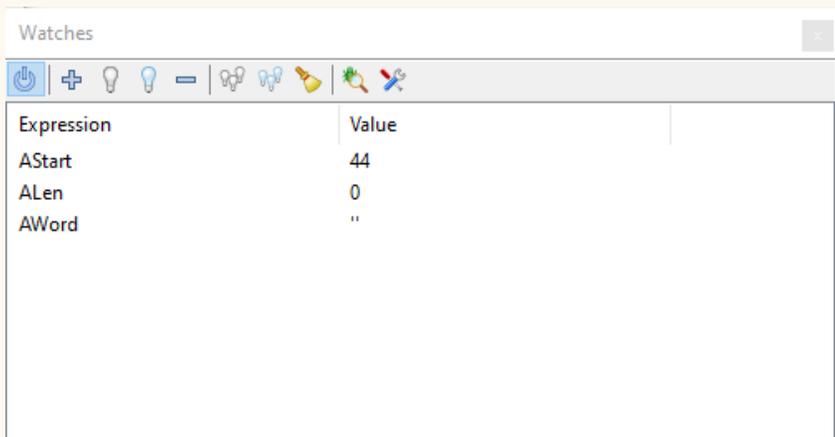


Abbildung 8

Wir können überprüfen, ob `AStart` korrekt ist, aber `ALen` ist es mit Sicherheit nicht. Wir haben 1 erwartet und 0 erhalten.

Damit können wir unsere Suche nach dem Fehler verfeinern. Wir wissen, dass `FindNextWord` die falsche Länge zurückgibt. Wir müssen in Zeile 27 einen Haltepunkt setzen, an dem wir die Quellwerte, aus denen die Länge berechnet wird, überprüfen können.

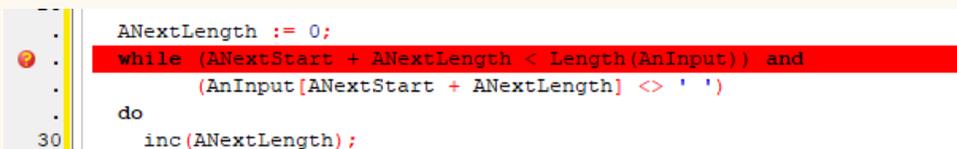


Abbildung 9

Bevor wir den neuen Haltepunkt setzen, starten wir das Programm neu und verwenden den vorhandenen Haltepunkt in Zeile 39, um das Programm so lange laufen zu lassen, bis uns das Überwachungsfenster anzeigt, dass das aktuelle "AWord" "B" ist.

Dann wissen wir, dass die nächste Iteration diejenige sein wird, bei der wir kein "C" erhalten.

Zu diesem Zeitpunkt wird der vorherige Haltepunkt in Zeile 39 nicht mehr benötigt.

Wenn wir ihn erneut mit der linken Maustaste anklicken, wird er entfernt.

Jetzt setzen wir auch den neuen Haltepunkt in Zeile 27 (siehe *Abbildung 9 auf dieser Seite*) und führen die Anwendung aus, um zu Zeile 27 zu gelangen.

Wir können diese Änderungen vornehmen, während die Anwendung ausgeführt wird, ohne sie neu kompilieren zu müssen. Und das ist ein Unterschied zur Verwendung von `writeln`.

Wir hätten kein `writeln` mitten in der Ausführung einfügen können.

Wenn wir den Haltepunkt in Zeile 27 erreichen, könnten wir die lokalen Variablen in das Überwachungsfenster einfügen.

Wir können aber auch einfach das Fenster Lokale Variablen mit `Strg-Alt-L` öffnen und es zeigt uns alle lokalen Variablen an.





Local Variables	
Name	Value
AnInput	'Just some random text ...
ACurrentWordStart	42
ANextStart	44
ANextLength	0
Sresult	True
FINDNEXTWORD	True
RESULT	True
AInputLength	44

Figure 10

Wir können sehen, dass `ANextStart=44`. Dies ist der Wert, der als Anfang von "C" zurückgegeben wird und ist derselbe, den wir zuvor im Uhrenfenster für `AStart` gesehen hatten. `ANextLength` ist immer noch Null, da es gerade erst initialisiert wurde. Sie muss einmal durch die Schleife erhöht werden, an deren Anfang wir pausiert haben. Die `while`-Bedingung muss also wahr sein und den Befehl `inc (ANextLength)` ausführen. Aber wir wissen bereits, dass sie Null zurückgeben wird. Fügen wir den ersten Teil der Bedingung zum Watches-Fenster hinzu. Markieren Sie den Text `"ANextStart + ANextLength < AinputLength"` im Editor und drücken Sie `Strg-F5`, um ihn zu Watches hinzuzufügen.

Watches	
Expression	Value
<code>ANextStart + ANextLength < AinputLength</code>	False

Abbildung 11

Und wir können sehen, dass es `false` zurückgibt (Abbildung 11 auf dieser Seite). Das ist nicht das, was es sein sollte. Wir haben also etwas gefunden, was falsch ist. Wenn wir zum Fenster Locals zurückgehen, sehen wir, dass nicht nur `ANextStart=44` ist, sondern auch `AnInputLength=44`. `AnextStart+ANextLength` kann also nicht kleiner als `AnInputLength` sein. Da wir auf das letzte Zeichen an Position 44 in der Zeichenkette zugreifen können, sollte die Bedingung nicht auf "kleiner als", sondern auf "kleiner oder gleich als" prüfen. Wir haben das Problem gefunden und können die Zeile so ändern, dass sie ein `"<="` enthält.

Restart line numbers at line 27

```

43.   while (ANextStart + ANextLength <= AInputLength) and
44.       (AnInput[ANextStart + ANextLength] <> ' ')
45.   do
46.       inc (ANextLength);
47.

```





ZUSAMMENFASSUNG

In diesem Artikel haben wir uns mit den folgenden Funktionen beschäftigt.

- Setzen und Aufheben von Haltepunkten
 - Mausklick im Zwischenraum
 - F5
 - Menü: Ausführen → Haltepunkt hinzufügen → Quellhaltepunkt
- Zum Haltepunkt laufen
 - Schaltfläche Ausführen
 - F9
 - Menü: Ausführen → Ausführen
- Hinweis verwenden, um Variablenwerte zu sehen
 - Bewegen der Maus
- Watches-Fenster verwenden, um Variablen und Ausdrucksergebnisse zu sehen
 - Strg-Alt-W
 - Menü: Ansicht → Debug-Fenster → Überwachen
 - [+]-Taste zum Hinzufügen von Beobachtungen
 - Strg-F5, um eine Editorauswahl oder ein Wort am Cursor hinzuzufügen
- Lokales Fenster verwenden.
 - Strg-Alt-L
 - Menü: Ansicht → Fenster debuggen → Lokale

Im nächsten Artikel werden wir uns mit dem "Steppen" befassen, mit dem wir das Programm Zeile für Zeile ausführen können.

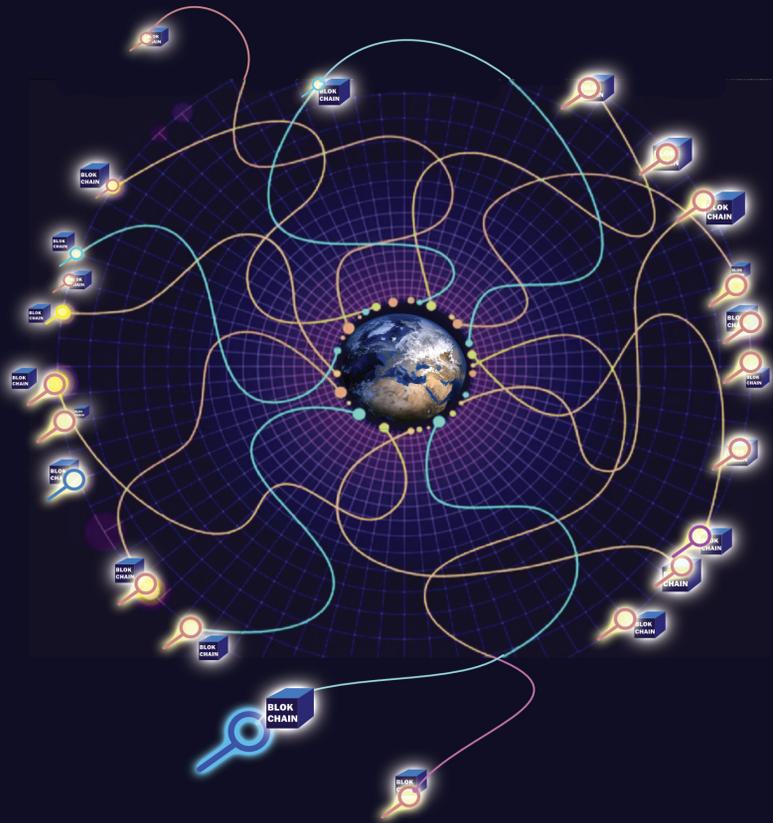


BLAISE PASCAL MAGAZINE 109

Multi platform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



Blaise Pascal



- English and German
- Free Lazarus PDF Kit Indexer
- 8 Issues per year
- minimal 60 pages
- Including example projects and code

€ 120 PER 2 YEAR

SUBSCRIPTION FOR 2 YEAR BLAISE PASCAL MAGAZINE



Starter

Expert



ZUSAMMENFASSUNG

In einem früheren Beitrag haben wir gezeigt, wie Sie einen Index von Wörtern in einer PDF-Datei erstellen können.

In diesem Artikel zeigen wir Ihnen, wie Sie diesen Index in einem Suchprogramm verwenden und damit eine Suche auf einer Website implementieren können.

1. EINLEITUNG

Im ersten Artikel über das Durchsuchen von PDF-Dateien haben wir gezeigt, wie man einen PDF-Indexer erstellt: ein Programm, das eine Reihe von PDF-Dateien analysiert und Vorkommen von Wörtern in den PDF-Dateien in einer Datenbank speichert.

All dies wurde mit Klassen erreicht, die Teil von Free Pascal sind:

Das Endergebnis war, dass wir eine Datenbank mit allen Wörtern in den PDF-Dateien und den Stellen, an denen diese Wörter vorkommen, haben.

In diesem Artikel zeigen wir, wie Sie diese Datenbank abfragen können:

Wir erstellen eine kleine Webseite, auf der der Benutzer einen Suchbegriff eingeben kann.

Der Suchbegriff wird an den Server gesendet, der daraufhin mit einer Liste von Seiten und den Titel des Artikels, zu dem die Seite gehört.

Der Benutzer kann dann auswählen, welchen Artikel er lesen möchte.

Die richtige PDF-Datei wird heruntergeladen und auf der richtigen Seite geöffnet.

Sobald die PDF heruntergeladen ist, kann der Benutzer lokal in dem PDF suchen. Der Einfachheit halber wird angenommen, dass sich die indizierten PDF-Dateien in einem Verzeichnis auf dem Server befinden, aber sie können sich natürlich auch in einer Datenbank befinden oder über viele Verzeichnisse verteilt sein.

Um dem Benutzer zu helfen, werden wir auch einen Autovervollständigungsmechanismus für das Suchfeld einrichten: Sobald der Benutzer 2 Buchstaben eingegeben hat, präsentiert das Programm eine Liste von Wörtern, die diese Buchstaben enthalten er oder sie eingegeben hat. Der Benutzer kann dann die vollständigen Wörter auswählen und das Backend danach suchen lassen. Das spart nicht nur Tipparbeit, sondern erhöht auch die Genauigkeit der Suche.

2 DER TEIL DER SERVERSUCHE: ÜBERSICHT

Der Server ist eine einfache HTTP-Anwendung, die 3 Funktionen hat:

- Geben Sie eine Liste von Wörtern aus der Datenbank zurück, basierend auf einigen Buchstaben des Wortes.
 - Liefert eine Liste von Übereinstimmungen für ein Wort, die den pdf-Namen, die Seiten mit einer Übereinstimmung und den Artikelnamen zu dem die Seite gehört.
 - Gibt eine PDF-Datei zurück.
- Um diese HTTP-Anwendung zu erstellen, starten wir eine neue HTTP Anwendung wie gewohnt in der Lazarus IDE.



Der Quellcode des Hauptprogramms wurde etwas geändert, so dass er wie folgt lautet:

```
begin
  ConfigureApp;
  {$ifndef usecgi}
  Application.Port:=3010;
  {$endif}
  Application.Initialize;
  Application.Run;
  ASearch.Free;
end.
```



Die `configureApp` ist ein Aufruf einer Routine, die den Suchmechanismus erstellt (die 2 ersten Anweisungen in der Routine) und den HTTP-Router konfiguriert, der für das Routing der 3 von der Anwendung unterstützten HTTP-Anfragen verantwortlich ist:

```
Var
  aSearch : TSearcher;
Procedure ConfigureApp;
begin
  // Create search mechanism
  aSearch:=TSearcher.Create(Application);
  aSearch.Init;
  // Register routes for the search mechanism
  HTTPRouter.RegisterRoute('/search',@aSearch.DocSearch,true);
  HTTPRouter.RegisterRoute('/list',@aSearch.WordList,False);
  // Set default file downloader
  DefaultFileModuleClass:=TCorsFileModule;
  // Where are the PDF files located.
  RegisterFileLocation('pdf',aSearch.PDFLocation);
end;
```

Die letzten 2 Zeilen verwenden einen Abkömmling eines Standardmoduls für den Dateidownload in Free Pascal (TFPCustomFileModule) namens `TCorsFileModule`, um eine Download-Funktion für die PDF-Dateien einzurichten. Beachten Sie, dass der Pfad natürlich übereinstimmen muss

Das `TCorsFileModule` ist dazu da, CORS zu aktivieren:

```
TCorsFileModule = Class(TFPCustomFileModule)
  Procedure HandleRequest(ARequest: TRequest; AResponse: TResponse); override;
end;

procedure TCorsFileModule.HandleRequest(ARequest: TRequest; AResponse: TResponse);
begin
  Cors.Enabled:=true;
  if Cors.HandleRequest(aRequest,aResponse) then exit;
  inherited HandleRequest(ARequest, AResponse);
end;
```

Das Standard-`TFPCustomFileModule` erlaubt keine CORS-Anfragen, daher wird ein Nachkomme erstellt, der dies ermöglicht. Es aktiviert die CORS-Unterstützung (Standardbestandteil jedes *FCL-Web Moduls*) und wenn eine CORS-Preflight-Anfrage erkannt wird, wird sie bearbeitet. Wenn es sich um eine normale Anfrage handelt, wird die PDF-Download-Anfrage bearbeitet: Mehr brauchen wir nicht zu tun, um die PDF-Dateien von einer URL zu liefern, die wie folgt aufgebaut ist:

`http://localhost:3010/pdf/somepdf.pdf`

(der Host und der Port können je nach Ihrer Einrichtung unterschiedlich sein) alles andere wird vom Dateimodul selbst erledigt.



Die eigentliche Suche erfolgt in der Klasse TSearcher, die wie folgt definiert ist:

```
TSearcher = Class(TComponent)
private
  procedure ConfigSearch(aRequest: TRequest; aResponse: TResponse);
  procedure ConfigWordList(aRequest: TRequest;
    Out aContaining : UTF8string;
    Out Partial : TAvailableMatch;
    Out aSimple : Boolean);

  procedure ConnectToDatabase;
  procedure DisconnectFromDatabase;
  function FindArticle(Position: Integer;
    const aFile: String): TArticleData;
  function SearchDataToJSON(aID: Integer;
    const aRes: TSearchWordData): TJSONObject;
  procedure SendJSON(J: TJSONObject; aResponse: TResponse);
  procedure SetupMetadata;
  procedure LoadArticles;
Protected
  function InitSearch(aResponse: TResponse): Boolean;
  function SetupDB(aIni: TCustomIniFile): TCustomIndexDB;
  Property DB : TCustomIndexDB;
  Property Search : TFPSearch;
  Property MinRank : Integer;
  Property FormattedJSON : Boolean;
  Property Articles : TArticleDataArray;
Public
  Procedure Init;
  procedure DoDBLog(Sender: TSQLConnection; EventType: TDBEventType;
    const Msg: String);
  Function CheckParams(aRequest : TRequest; aResponse : TResponse) : Boolean;
  Function CheckSearchParams(aRequest : TRequest; aResponse : TResponse) : Boolean;
  Procedure DocSearch(aRequest : TRequest; aResponse : TResponse);
  Procedure WordList(aRequest : TRequest; aResponse : TResponse);
  Property AllowCors : Boolean;
end;
```



Die Routinen DocSearch und WordList sind die Einstiegspunkte für die Bearbeitung der Browseranfragen, wir werden sie gleich behandeln. Die Init-Routine, die beim Programmstart aufgerufen wird, initialisiert die Suchmaschine. Sie liest eine Konfigurationsdatei mit den Einstellungen für die Datenbankverbindung und einigen weiteren Einstellungen, die den Suchmechanismus konfigurieren. Die Konfigurationsdatei wird in mehreren Verzeichnissen durchsucht, in der Routine GetConfigFileName, die wir hier nicht vorstellen werden.

```
procedure TSearcher.Init;
Const
  // Adapt this default to your setup...
  DefaultLocation = '/home/michael/Documents/pdf/blaise/';
Var CFN : String; aIni: TMemIniFile;
begin
  CFN:=GetConfigFileName;
  aIni:=TMemIniFile.Create(CFN);
  try
    FFormattedJSON:=aIni.ReadBool('search','formatjson',False);
    FDefaultMinRank:=aIni.ReadInteger('search','minrank',1);
    FDefaultMetadata:=aIni.ReadBool('search','metadata',true);
    FAllowCors:=aIni.ReadBool('search','allowcors',true);
    FPDFLocation:=aIni.ReadString('Files','PDF',DefaultLocation);
    FDB:=SetupDB(aIni);
    FSearch:=TFPSearch.Create(Self);
    FSearch.Database:=FDB;
  finally
    aIni.Free;
  end;
  SetupMetadata;
end;
```



4 Variablen aus der Konfigurationsdatei initialisiert werden, beeinflussen sie die zurückgegebenen Ergebnisse auf die Client-Anfragen:

- FFormattedJSON** Soll das zurückgegebene JSON formatiert sein oder nicht.
- FDefaultMinRank** Mindestrang für ein Wortvorkommen, um in das in das Ergebnis aufgenommen wird.
- FDefaultMetadata** Sollen Metadaten zurückgegeben werden, wenn kein Parameter zurückgegeben werden, der sie angegeben hat.
- FAllowCors** Sollen CORS HTTP-Header für das Ergebnis gesetzt werden?
- FPDFLocation** Der Speicherort der PDF-Dateien auf der Festplatte.



Die **FDB** ist eine Instanz von TCustomIndexDB, wie im vorigen Artikel beschrieben: Sie verwaltet die Datenbankverbindung für die TFPSearch-Instanz, die die eigentliche Suche in der Datenbank durchführt.

Die Klasse TFPSearch ist Teil der im vorigen Artikel vorgestellten Einheit fpIndexer. Sie verwaltet die Suche in der Indexdatenbank. Die Klasse wird wie folgt deklariert:

```
TFPSearch = class (TComponent)
public
  constructor Create(AOwner: TComponent); override;
  destructor Destroy; override;
  function Execute: int64;
  procedure AddResult(index: integer; AValue: TSearchWordData);
  procedure SetSearchWord(AValue: UTF8String);
  function GetAvailableWords(out aList : TUTF8StringArray;
    aContaining : UTF8String;
    Partial : TAvailableMatch) : Integer;

  property Count: integer;
  property RankedCount: integer;
  property Results[index: integer]: TSearchWordData ;
  property RankedResults[index: integer]: TSearchWordData;
published
  property Database: TCustomIndexDB;
  property Options: TSearchOptions;
  property SearchWord: TWordParser;
  Property UsePositionInRank : Boolean;
end;
```

Die veröffentlichten Eigenschaften können verwendet werden, um den Suchmechanismus zu konfigurieren:

Die Eigenschaft Database stellt die Verbindung zur Datenbank dar.

Die Optionen können auf soContains gesetzt werden. In diesem Fall wird die Suche nach Wörtern, die den Suchbegriff enthalten.

UsePositionInRank legt fest, ob die Position bei der Bewertung von Wörtern berücksichtigt werden soll (normalerweise wird nur die Datei/der Name berücksichtigt)

Die Methoden der Klasse haben den folgenden Zweck:

- SetSearchWord Legen Sie das Wort fest, nach dem gesucht werden soll.
- Execute Starten Sie die Suche. Gibt die Anzahl der gefundenen Ergebnisse zurück.
- AddResult Fügt eine Übereinstimmung zum Ergebnisindex hinzu.
- GetAvailableWords Ermittelt die Liste der Wörter, die das Wort aEnthalten.
Der Abgleichmechanismus wird durch den Wert des des Parameters Partial bestimmt: einer von
amAll (alle Wörter),
amExact (exakte Übereinstimmung mit dem Suchbegriff),
amContains (Wörter, die den Suchbegriff enthalten),
amStartsWith (Wörter, die mit dem Suchbegriff beginnen).

Die Ergebnisse werden in dem Array aList zurückgegeben und die Funktion gibt die Anzahl der Ergebnisse zurück.

Die öffentlichen Eigenschaften der Klasse ermöglichen es, die Ergebnisse der Suche zu untersuchen.

Die Eigenschaften Results und Count beschreiben die Rohergebnisse des Suchvorgangs.



Die letzte Zeile der `Init`-Routine ist ein Aufruf von `SetupMetadata`. Diese Routine richtet Metadaten Definitionen für die 2 REST-Ergebnismengen ein, die vom Serverprogramm verarbeitet werden können: Das Programm ist so eingerichtet, dass das von ihm zurückgegebene Ergebnis eine JSON-Struktur ist, die von der `TJSONDataset`-Implementierung in `pas2js` konsumiert werden kann.



Die JSON-Struktur, die von einem `TJSONDataset`-Datensatz konsumiert wird, enthält ein Element "Metadaten", das die Daten beschreibt. Anstatt diese Struktur jedes Mal zu erstellen, wenn wir eine Ergebnismenge erstellen, erstellen wir sie einmal und verwenden sie wieder, wenn ein Ergebnis zurückgegeben wird.

```
procedure TSearcher.SetupMetadata;  
begin  
  FMetadata:=TJSONObject.Create([  
    'root','data',  
    'idField','id',  
    'fields',TJSONArray.Create([  
      TJSONObject.Create(['name','id','type','int']),  
      TJSONObject.Create(['name','rank','type','int']),  
      TJSONObject.Create(['name','articlePage','type','int']),  
      TJSONObject.Create(['name','articleIssue',  
        'type','string','maxlen',10]),  
      TJSONObject.Create(['name','articleAuthor',  
        'type','string','maxlen',127]),  
      TJSONObject.Create(['name','articleTitle',  
        'type','string','maxlen',255]),  
      TJSONObject.Create(['name','url',  
        'type','string','maxlen',100]),  
      TJSONObject.Create(['name','context',  
        'type','string',  
        'maxlen',MaxContextLen]),  
      TJSONObject.Create(['name','date','type','date'])  
    ])  
  ]);  
  FWordsMetadata:=TJSONObject.Create([  
    'root','data',  
    'idField','id',  
    'fields',TJSONArray.Create([  
      TJSONObject.Create(['name','id','type','int']),  
      TJSONObject.Create(['name','word',  
        'type','string','maxlen',100])  
    ])  
  ]);  
end;
```

3 THE SERVER SEARCH PART: THE WORD LIST FOR AUTO-COMPLETION

Die `WordList`-Routine wird immer dann aufgerufen, wenn der Benutzer einen Suchbegriff eingibt und der Browser eine Vervollständigungsliste anzeigen möchte.

Die `WordList`-Ressource akzeptiert mehrere Abfrageparameter:

- q** (für `query`), dieser obligatorische Parameter enthält die vom Benutzer eingegebenen Buchstaben.
- t** (für `Typ`) bestimmt, welche Wörter zurückgegeben werden: Der Wert ist einer der folgenden
 - `all` gibt alle Wörter zurück, der Wert von `q` muss leer sein.
 - `contains` gibt alle Wörter zurück, die den Wert von `q` enthalten.
 - `exact` gibt nur eine exakte Übereinstimmung zurück: so können Sie feststellen, ob das Wort in der Datenbank vorhanden ist.
 - `starts with` gibt alle Wörter zurück, die mit dem Wert von `q` beginnen.
- s** ein boolescher Wert (0 oder 1), wenn `True`, dann enthält die Rückgabe nur ein Array von Wörtern. Wenn `False`, dann wird auch ein `id`-Feld zurückgegeben.
- m** ein boolescher Wert (0 oder 1): schließt Metadaten in die Antwort ein oder nicht. Wenn Sie nichts angeben, wird der Standardwert aus der Konfigurationsdatei verwendet.



Wenn sie aufgerufen wird, beginnt sie mit der Initialisierung der Antwort, die an den Browser gesendet wird, einschließlich der CORS-Header für die optionale CORS-Unterstützung, und fährt dann mit der Überprüfung der Abfrageparameter fort: Die CheckSearchParams-Routine prüft, ob die angegebenen Abfrageparameter gültig sind.

Wenn nicht, wird ein Fehlercode in die HTTP-Anfrage eingefügt und False zurückgegeben.

Sie wird hier nicht vorgestellt. Die InitSearch-Routine prüft lediglich, ob die Datenbank verfügbar ist.



```

procedure TSearcher.WordList(aRequest: TRequest; aResponse: TResponse);
Var
    I : Integer;
    J : TJSONObject;
    A : TJSONArray;
    w, aContaining      : UTF8String;
    aPartial            : TAvailableMatch;
    aSimple             : Boolean;
    aList               : TUTF8StringArray;
begin
    aResponse.ContentType := 'application/json';
    if AllowCORS then
        AResponse.SetCustomHeader('Access-Control-Allow-Origin','*');
    if not CheckSearchParams(aRequest,aResponse) then exit;
    if not InitSearch(aResponse) then exit;
    ConfigWordList(aRequest,aContaining,aPartial,aSimple);
    FSearch.GetAvailableWords(aList,aContaining,aPartial);
    J:=TJSONObject.Create;
    try
        if FIncludeMetadata then J.Add('metaData',FWordsMetadata.Clone);
        A:=TJSONArray.Create;
        if aSimple then
            For W in aList do A.Add(W)
        else
            begin
                For I:=0 to Length(aList)-1 do
                    A.Add(TJSONObject.Create(['id',I+1,'word',aList[i]]));
            end;
            J.Add('data',A);
            SendJSON(J,aResponse);
        finally
            J.Free;
        end;
    end;

```

Die Routine ConfigWordList extrahiert die Abfrageparameter aus der Anfrage und stellt sicher, dass die zurückgegebenen Werte (aContaining, aPartial, aSimple) konsistent sind.

Man könnte diese Funktionalität in die Routine CheckSearchParams integrieren.

Die Funktion FSearch.GetAvailableWords wird verwendet, um die eigentlichen Ergebnisse zu erhalten.

Sobald die Ergebnisse vorliegen, besteht der Rest der Routine nur noch darin, das JSON zu konstruieren, das an den Browser zurückgegeben wird: Hier wird der Parameter 'simple' verwendet, um die Struktur der Elemente im Ergebnis-Array zu bestimmen.

SendJSON gibt einfach das konstruierte JSON in der HTTP-Antwort zurück:

```

procedure TSearcher.SendJSON(J : TJSONObject; aResponse: TResponse);
begin
    if FormattedJSON then
        aResponse.Content:=J.FormatJSON()
    else
        aResponse.Content:=J.AsJSON;
        aResponse.ContentLength:=Length(aResponse.Content);
        aResponse.SendContent;
    end

```



Damit ist die Funktionalität, eine Wortliste zu erhalten, vollständig. Da es sich um eine einfache HTTP-Anfrage handelt, kann sie einfach im Browser getestet werden, indem Sie z.B. die folgende URL verwenden:

`http://localhost:3010/list?q=class&t=contains&s=1`

Ein Beispielergebnis für die Wortklasse sehen Sie in Abbildung 1 auf Seite 7 dieses Artikels. Beachten Sie, dass der Browser anhand des Inhaltstyps erkennt, dass es sich um JSON handelt, und das Ergebnis entsprechend formatiert.



4 DER TEIL DER SERVERSUCHE: DIE LISTE DER SEITEN MIT WORTÜBEREINSTIMMUNGEN

Der zweite Aufruf, der im Server benötigt wird, findet alle Vorkommen eines Wortes in den indizierten PDFs. Der PDF-Indexer speichert für jedes Wort den Namen der Datei, in der das Wort vorkommt, und die Seite, auf der das Wort vorkommt.

Für die Anwendung, die wir machen wollen, muss die Seitenzahl abgeglichen werden mit einem Artikel abgeglichen werden: Um eine Seite einem Artikel zuzuordnen, haben wir eine Tabelle mit dem PDF-Namen (und der Ausgabe), den Anfangs- und Endseiten, dem Autor und dem Titel. Da es nur eine begrenzte Anzahl von Artikeln gibt und sich die Liste nicht so oft ändert, ist es sinnvoll, die Liste auf dem Server zwischenzuspeichern. Wir tun dies in dem folgenden Array von Datensätzen:

```
TArticleData = record
  StartPage,
  EndPage : Integer;
  Author,
  Title,
  Issue,
  PDF : String;
  function Match(aPDF : String; aPage : Integer) : boolean;
end;
TArticleDataArray = Array of TArticleData;
```

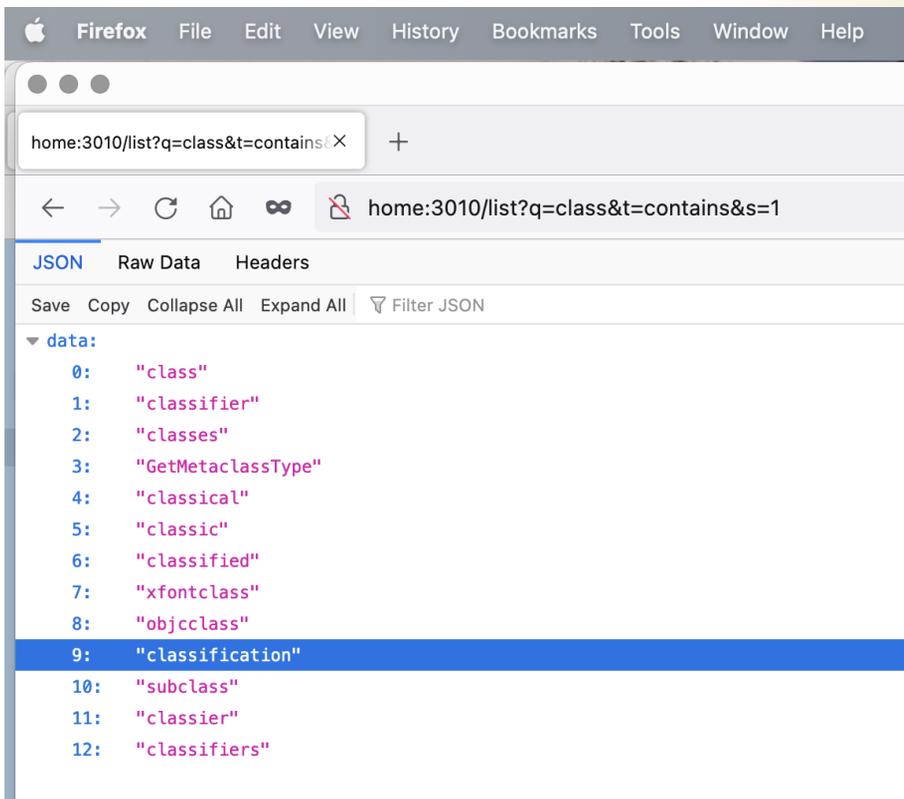


Abbildung 1: Das Ergebnis einer Wortlistenabfrage



Diese Informationen werden mit dem Ergebnis der PDF-Indexdaten abgeglichen. Der HTTP-Einstiegspunkt für den Suchmechanismus ist der Pfad 'search', der mit der Methode PDFSearch verknüpft ist. Ähnlich wie der WordList-Einstiegspunkt akzeptiert die PDFSearch-Methode einige Optionen über die Abfragevariablen:



- q** (für query), dieser obligatorische Parameter enthält das Wort, nach dem gesucht werden soll.
- r** (für rank) bestimmt den minimalen Rang, den eine Übereinstimmung haben muss, damit die Übereinstimmung in das Ergebnis aufgenommen wird. Der Standardwert ist Null.
- c** ein boolescher Wert (0 oder 1), wenn True, werden Wörter zurückgegeben, die die Abfrage enthalten. Wenn False (die Voreinstellung), wird nur eine exakte Wortübereinstimmung zurückgegeben.
- m** ein boolescher Wert (0 oder 1): schließt Metadaten in die Antwort ein oder nicht. Wenn Sie es weglassen, wird die Standardwert wie in der Konfigurationsdatei verwendet.

Die PDFSearch-Methode ist überraschend kurz und hat dieselbe Gesamtstruktur wie die WordList-Methode. Sie beginnt damit, die Antwort vorzubereiten, CORS-Header zu setzen und die Parameter zu überprüfen. Der nächste Schritt ist die Initialisierung und Konfiguration der Suche. Wenn die Initialisierung schief geht, wird die Routine beendet.

```
procedure TSearcher.PDFSearch(aRequest: TRequest; aResponse: TResponse);
Var
  I : Integer;
  J : TJSONObject;
  A : TJSONArray;
begin
  aResponse.ContentType:='application/json';
  if AllowCORS then
    AResponse.SetCustomHeader('Access-Control-Allow-Origin','*');
  if not CheckParams(aRequest,aResponse) then exit;
  if not InitSearch(aResponse) then
    begin
      aResponse.Code:=500;
      aResponse.CodeText:='Internal error';
      aResponse.SendResponse;
      exit;
    end;
  ConfigSearch(aRequest,aResponse);
```

Die ConfigSearch konfiguriert das FPC-Index-Suchobjekt anhand der Abfragevariablen:

```
procedure TSearcher.ConfigSearch(aRequest: TRequest; aResponse: TResponse);
Var
  S : string;
  O : TSearchOptions;
  B : Boolean;
begin
  FMinRank:=StrToIntDef(aRequest.QueryFields.Values['r'],0);
  if FMinRank=0 then
    FMinRank:=FDefaultMinRank;
  S:=aRequest.QueryFields.Values['m'];
  if (S="") or Not TryStrToBool(S,FIncludeMetaData) then
    FIncludeMetaData:=FDefaultMetaData;
  FSearch.SetSearchWord(aRequest.QueryFields.Values['q']);
  O:=[];
  S:=aRequest.QueryFields.Values['c'];
  if (S<>"") and TryStrToBool(S,B) and B then
    Include(O,soContains);
  FSearch.Options:=O;
end;
```

Wenn der Aufruf der Routine ConfigSearch den Suchmechanismus konfiguriert hat, startet die PDFSearch-Routine die eigentliche Suche: Der Aufruf von FSearch.Execute führt die erforderlichen SQL-Anweisungen in der Indexdatenbank aus. Die Ergebnisse sind in den Array-Eigenschaften Results und RankedResults verfügbar. Letztere werden in der Routine SearchDataToJSON mit dem Array articles kombiniert, um die eigentliche Ergebnismenge zu bilden:



```
FSearch.Execute;  
A:=nil;  
J:=TJSONObject.Create;  
try  
  if FIncludeMetadata then  
    J.Add('metaData',FMetadata.Clone);  
  A:=TJSONArray.Create;  
  For I:=0 to Search.RankedCount-1 do  
    begin  
      if Search.RankedResults[I].Rank>=MinRank then  
        A.Add(SearchDataToJSON(I+1,Search.RankedResults[I]));  
    end;  
  J.Add('data',A);  
  SendJSON(J,aResponse);  
finally  
  J.Free;  
end;
```



Das Ergebnis wird bei Bedarf mit den Metadaten kombiniert und mit dem Befehl SendJSON an den Browser gesendet, genau wie in der Routine WordList. Die Suchdaten werden mit der Artikelliste in der Routine SearchDataToJSON kombiniert, die damit beginnt, den Artikel zu finden, der mit der Seite der Wortübereinstimmung (gespeichert in Position) übereinstimmt. Der gefundene Artikel und das Suchergebnis werden dann in einer einzigen JSON-Struktur vereinigt:

```
function TSearcher.SearchDataToJSON(aID: Integer; const aRes: TSearchWordData): TJSONObject;  
Var  
  Article : TArticleData;  
begin  
  Article:=FindArticle(aRes.Position,aRes.URL);  
  Result:=TJSONObject.Create([  
    'id',aID,  
    'rank',aRes.Rank,  
    'articlePage', aRes.Position,  
    'articleTitle', Article.Title,  
    'articleAuthor', Article.Author,  
    'articleIssue', Article.Issue,  
    'url',aRes.URL,  
    'context',ares.Context,  
    'date',FormatDateTime('yyyy'-'mm'-'dd'T'hh':'nn':'ss',aRes.FileDate)  
  ]);  
end
```

Die Routine, um die Artikel zu finden, ist denkbar einfach:

```
function TSearcher.FindArticle(Position: Integer; const aFile: String): TArticleData;  
Var  
  A : TArticleData;  
begin  
  Result:=Default(TArticleData);  
  if Length(Articles)=0 then  
    LoadArticles;  
  for A in Articles do  
    if A.Match(aFile,Position) then  
      Exit(A);  
end;
```

Die LoadArticles Routine führt eine einfache Select-SQL-Abfrage aus und speichert die Ergebnisse im Array Articles, das Teil der Klasse TSearcher ist. Der interessierte Leser kann den mit den Artikeln gelieferten Code konsultieren, um die Details zu erfahren.

Die Match-Methode des TArticleData-Datensatzes gibt True zurück, wenn sich der Artikel in der PDF-Datei befindet und die Seitenzahl enthält:



```
function TArticleData.Match(aPDF : String; aPage : Integer) : boolean;
begin
  Result := SameText(Pdf, aPdf)
    and (StartPage <= aPage)
    and (aPage <= EndPage)
end;
```



Mit dieser Routine haben wir den Suchmechanismus in der Indexdatenbank abgedeckt. Auch hier kann das Ergebnis einfach im Browser getestet werden, wie in Abbildung 2 auf Seite 10 dieses Artikels gezeigt.

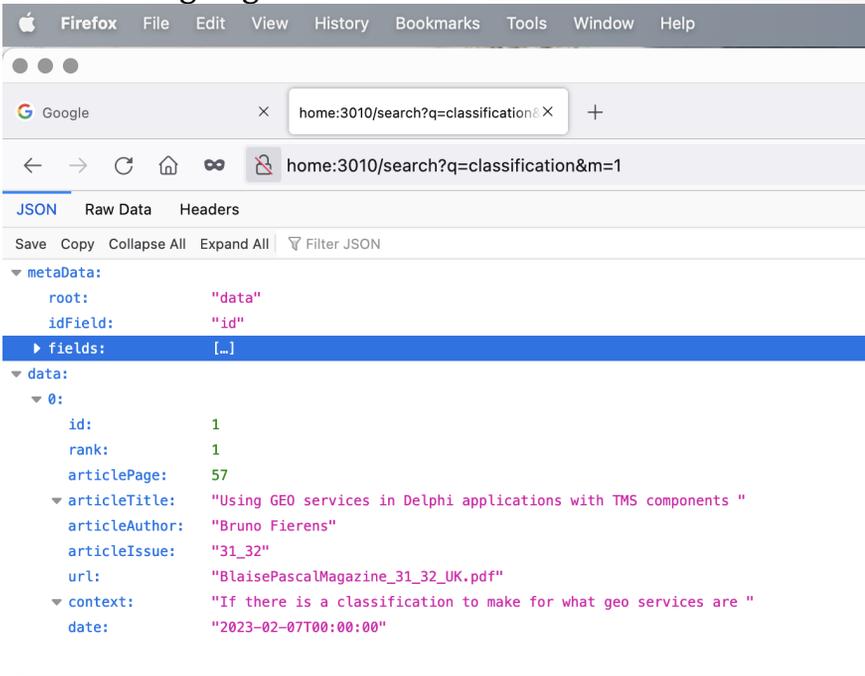


Abbildung 2: Das Ergebnis einer Suchanfrage

5 THE BROWSER APPLICATION

Da wir nun eine API haben, mit der wir Wortlisten und Übereinstimmungen eines Wortes in unserer Sammlung von PDF-Dokumenten haben, können wir eine Browser-Anwendung erstellen, die diese APIs verwendet, um eine Sammlung von PDF-Dokumenten abzufragen und anzuzeigen.

Dazu werden wir die in einem früheren Beitrag vorgestellte PDF-Anzeigeapplication überarbeiten Beitrag, Ausgabe 105 Seite 27, vorgestellten PDF-Anwendung überarbeiten und sie so ändern, dass wir entweder lokal in der angezeigten PDF-Datei oder in der PDF-Sammlung auf dem Server suchen können.

Wir beginnen mit dem Refactoring der Anwendung: Ursprünglich bestand die Anwendung aus aus einer einzigen Anwendungsklasse. Diese wird nun in 4 verschiedene Klassen aufgeteilt: TPDFPanel diese Klasse abstrahiert den PDF-Viewer PDF.js. Sie hat eine Methode zum Laden einer PDF-Datei, eine Methode zum Einstellen der aktuellen Seite und 2 Ereignisse, die aufgerufen werden aufgerufen werden, wenn die PDF-Datei geladen bzw. die Seite angezeigt wird.

```
TPDFPanel = Class(TComponent)
  Procedure QueueRenderPage(aNum : Integer);
  Procedure ShowPDF(aSource : TPDFSource; AtPage : Integer = 1);
  Property CanvasID : String;
  Property PageRendering : Boolean;
  Property Scale : Double;
  Property DisplayedPage : Integer;
  Property PageCount : Integer;
  Property OnPageLoaded : TNotifyEvent;
  Property OnLoaded : TNotifyEvent;
  Property PDFDoc : TPDFDocumentProxy;
end;
```



Die Eigenschaften sind ziemlich einfach: Die CanvasID kann auf die ID des Canvas-Elements gesetzt werden, in dem das PDF-Element gezeichnet wird. Die Eigenschaft Scale dient dazu, den Zoomfaktor für die PDF-Datei festzulegen. Die schreibgeschützten Eigenschaften PageRendering, DisplayedPage, PageCount und PDFDoc haben offensichtliche Bedeutungen.

TPDFControlPanel Diese Klasse enthält die Steuerelemente für den PDF-Viewer: Die Schaltflächen für die Seitennavigation und die Zoom-Schaltfläche. Die Komponente hat eigentlich nur einige öffentliche Eigenschaften zum Einstellen des PDF-Panels und der IDs der verschiedenen HTML-Elemente.



```
TPDFControlPanel = class(TComponent)
  Procedure BindElements;
  Property PDFPanel : TPDFPanel Read FPDFPanel Write SetPDFPanel;
  Property EditPageID      : string;
  Property PreviousButtonID : string;
  Property ZoomLabelID     : string;
  // Properties for all other buttons
end;
```

Der BindElements-Aufruf holt die Verweise auf alle benötigten Elemente in der HTML-Seite. TPDFSearchControl Diese Klasse kümmert sich um den eigentlichen Suchmechanismus: Sie hat einige Eigenschaften für die verschiedenen IDs der HTML-Tags, die sie für ihre Arbeit benötigt und einige Methoden, um das Suchergebnis-Panel anzuzeigen/auszublenden/zu löschen.

```
TPDFSearchControl = class(TComponent)
  procedure BindElements;
  Procedure ShowResultPanel;
  Procedure HideResultPanel;
  Procedure ClearResultPanel;
  Property pdfPanel : TPDFPanel;
  Property LocalCheckboxID : String;
  Property SearchButtonID : String;
  Property SearchEditID : String;
  Property DivAutoCompleteID : String;
  Property ResultPanelID : String;
  Property SidebarPanelID : String;
  Property ShowResultsPanelButtonID;
end;
```

Auch hier holt der BindElements-Aufruf die Verweise auf alle benötigten Elemente in der HTML-Seite. Den Code der Klassen TPDFControlPanel und TPDFPanel finden Sie im vorigen Artikel über die Anzeige einer PDF-Datei, daher wird er hier nicht wiederholt. Der einzige geänderte Code besteht darin, dass die Klasse TPDFControlPanel nicht auf die Low-Level-Klasse TPdfDoc zugreift, sondern die Methoden von TPDFPanel verwendet (die ihrerseits natürlich die Methoden der Klasse TPdfDoc aufrufen).

Die Anwendungsklasse TPDFIndexApp verbindet alles miteinander: Sie hat Instanzen der der drei oben genannten Klassen und einige Verweise auf die HTML-Elemente zum Laden einer PDF-Datei von der Festplatte:

```
TPDFIndexApp = class(TBrowserApplication)
  pnlSidebar,
  lblFileLocation : TJSHTMLElement;
  btnLoad,
  btnClosePane : TJSHTMLButtonElement;
  edtPDFFile : TJSHTMLInputElement;
  FPDFPanel : TPDFPanel;
  FPDFControls : TPDFControlPanel;
  FPDFSearch : TPDFSearchControl;
  procedure doRun; override;
  procedure BindElements;
  function DoLoadFile(Event: TEventListenerEvent): boolean;
  procedure onLoad(aEvent: TJSEvent);
  procedure ShowPDF(aSource: TPDFSource);
  Procedure DisplayFileLocation(const aLocation : String);
  procedure onClosePane(aEvent: TJSEvent);
end;
```



Die verschiedenen Ereignishandler in der Anwendungsklasse werden hier nicht behandelt, der Code hat sich im Vergleich zu dem in dem Artikel über das Anzeigen einer PDF-Datei vorgestellten Code nicht geändert. Der DoRun-Code hat sich geändert, daher stellen wir ihn hier vor:



```
procedure TPDFIndexApp.DoRun;  
const  
  TheURL = 'https://mozilla.github.io/pdf.js/build/pdf.worker.js';  
begin  
  pdfjsLib.GlobalWorkerOptions.workerSrc:=TheURL;  
  FPDFPanel := TPDFPanel.Create(Self);  
  FPDFControls := TPDFControlPanel.Create(Self);  
  FPDFControls.PDFPanel := FPDFPanel;  
  FPDFSearch := TPDFSearchControl.Create(Self);  
  FPDFSearch.PDFPanel := FPDFPanel;  
  BindElements;  
  Terminate;  
end;
```

Wie Sie im Code sehen können, beschränkt sich der Code auf die Erstellung der verschiedenen Panels und die Übergabe eines Verweises auf die TPDFPanel-Instanz an das TPDFControls-Panel. Der BindElements-Code ist stark reduziert, da die meisten Elemente jetzt in die in die anderen Klassen verschoben wurden. Deren BindElements-Methode wird von der Anwendungsklasse BindElements-Methode aufgerufen.

```
procedure TPDFIndexApp.BindElements;  
begin  
  btnLoad := TJSHTMLButtonElement(GetHTMLInputElement('btnLoad'));  
  btnLoad.addEventListener('click', @onLoad);  
  lblFileLocation := GetHTMLInputElement('lblFileLocation');  
  edtPDFFile := TJSHTMLInputElement(GetHTMLInputElement('edtPDFFile'));  
  edtPDFFile.onChange := @DoLoadFile;  
  btnCloseSidebar := TJSHTMLButtonElement(GetHTMLInputElement('btnClosePane'));  
  btnCloseSidebar.addEventListener('click', @onClosePane);  
  pnlSidebar := TJSHTMLInputElement(GetHTMLInputElement('pnlSidebar'));  
  FPDFPanel.CanvasID := 'PDFCanvas'; // Will call bindelements  
  FPDFControls.BindElements;  
  FPDFSearch.BindElements  
end;
```

Die anderen Elemente und Ereignisse beziehen sich auf das Laden einer vom Benutzer ausgewählten PDF-Datei: die ursprüngliche Funktionalität, eine beliebige Datei suchen zu können, bleibt erhalten. Die Klassen TPDFPanel und TPDFControlPanel enthalten keinen Code, der sich vom Artikel über das Anzeigen einer PDF-Datei unterscheidet, daher wird ihr Code hier nicht wiederholt. Das TPDFSearchControl ändert sich, da nun 2 Suchmechanismen unterstützt werden müssen, sowie ein Mechanismus zur Anzeige einer Autovervollständigungsliste. Wir beginnen mit der Autovervollständigungsliste. Sie wird im on input-Ereignis des edit-Elements aktiviert. Der Event-Handler wird in der BindElements-Methode festgelegt:

```
edtSearch := TJSHTMLInputElement(GetHTMLInputElement(SearchEditID));  
edtSearch.onkeyup := @DoSearchKeyUp;  
edtSearch.oninput := @DoCompleteWord;
```

Die Methode DoCompleteWord ist dafür verantwortlich, einen Timer (200 ms) zu starten (oder neu zu starten), und wenn der Timer abläuft, wird die Wortvervollständigungsliste abgerufen. Das Abrufen der Wortvervollständigungsliste erfolgt durch das Öffnen eines Datensatzes:

```
function TPDFSearchControl.DoCompleteWord(Event: TEventListenerEvent): boolean;
```

```
  procedure DoServerSearchWord;  
  begin  
    if Length(edtSearch.Value)>1 then  
      begin  
        FSearchTerm:=edtSearch.Value;  
        FWords.Close;  
        FWords.Load([],Nil);  
      end;  
    end;  
  end;  
begin  
  Result:=False;  
  if FSearchTimerID<>0 then  
    window.clearTimeout(FSearchTimerID);  
  FSearchTimerID:=window.SetTimeout(@DoServerSearchWord,200);  
end;
```



Wie im ersten Teil dieses Artikels erwähnt, kann das Ergebnis der Serveraufrufe in einem Format vorliegen, das für einen Restdatensatz geeignet ist. Die `FWords`-Variable in dem obigen Code ist ein `TRestDataset` und wird in der Methode `SetupDatasets` der Klasse `TPDFSearchPanel` Klasse initialisiert:



```
procedure TPDFSearchControl.SetupDatasets;  
Const  
    ServerURL = 'http://localhost:3010/';  
begin  
    FConn := TRESTConnection.Create(Self);  
    FConn.BaseURL := ServerURL;  
    FConn.OnGetURL := @DogetURL;  
    FResult := TRestDataset.Create(Self);  
    FResult.Connection := FConn;  
    FResult.AfterOpen := @DoOpenResults;  
    FWords := TRestDataset.Create(Self);  
    FWords.Connection := FConn;  
    FWords.AfterOpen := @DoWordsOpen;  
end;
```

Die Methode richtet auch den `FResult`-Datensatz ein, der später das Ergebnis der Suche nach Wortübereinstimmungen später verarbeiten wird. `FConn.OnGetURL` ist ein Ereignis, das von der Komponente `TRESTConnection` aufgerufen wird. Es kann dazu verwendet werden, die URL für einen Datensatz zu erstellen, der seine Daten von einem REST Server bezieht. In unserer Implementierung müssen wir den richtigen Endpunkt wählen, je nachdem welches Dataset geöffnet wird:

```
procedure TPDFSearchControl.DogetURL(Sender: TComponent; aRequest: TDataRequest;  
var aURL: String);  
var  
    Q: String;  
begin  
    Q := encodeURIComponent(FSearchTerm);  
    if aRequest.Dataset = FResult then  
        aURL := FConn.BaseURL + 'search?m=1&q=' + Q  
    else  
        aURL := FConn.BaseURL + 'list?t=contains&m=1&q=' + Q;  
end;
```

Sie sehen oben, dass die 2 Endpunkte, die wir im Server definiert haben, für beide Datensätze verwendet werden. In beiden Fällen sind die Abfragevariablen auf die Abfrage von Metadaten eingestellt, und `q` wird auf die Variable `FSearchTerm` gesetzt, die der Wert des `edtSearch` HTML-Eingabe-Tag ist. Das Ereignis `AfterOpen` der Klasse `TDataset` wird verwendet, um den HTML-Code für die Wortvervollständigungsliste. Der HTML-Code für diese Vervollständigungsliste ist eigentlich recht einfach. Die Bearbeitungssteuerung ist in 2 Klassen verpackt:

```
<p class="control is-small">  
    <div class="dropdown">  
        <div class="dropdown-trigger">  
            <input id="edtSearch" class="input" style="max-width: 15em;">  
            <div class="dropdown-menu" id="mnuAutoComplete" role="menu" />  
        </div> <!-- .dropdown-trigger -->  
    </div> <!-- .dropdown -->  
</p>
```

Und das HTML für die Vervollständigungsliste wird unterhalb des Tags mit der ID `mnuAutoComplete` eingefügt.

Sie wird erzeugt, indem die Datensätze im Datensatz in einer Schleife durchlaufen und die erforderlichen HTML-Elemente erzeugt werden. Die Routine beginnt damit, das Element auszublenden, den Inhalt zu löschen und dann die Elemente hinzuzufügen:



```
procedure TPDFSearchControl.DoWordsOpen(DataSet: TDataSet);
Var S : String;
    F : TField;
    P : TJSHTMLElement;
    A : TJSHTMLAnchorElement;
begin
  mnuAutoComplete.style.setProperty('display','none');
  mnuAutoComplete.InnerHTML:='

Beachten Sie, dass das Ereignis 'OnClick' für jedes Wort gesetzt wird. Am Ende wird das Menü wieder sichtbar gemacht. Abbildung 3 auf Seite 14 dieses Artikels zeigt, wie sich dieser Code auswirkt. Natürlich lässt sich die Liste mit etwas CSS noch viel schöner gestalten. Der OnClick-Ereignishandler für die Wörter setzt das Wort in die Suchbearbeitung, was ein wirklich einfacher Vorgang ist:



Abbildung 3: Die Liste zum automatischen Vervollständigen



```
procedure TPDFSearchControl.DoWordSelected(Event: TJSEvent);
begin
 event.PreventDefault;
 edtSearch.value:=event.targetelement.innerText;
 mnuAutoComplete.style.setProperty('display','none');
end;
```



Wenn der Benutzer in der Sucheingabe die Eingabetaste drückt oder auf den Button Suche klickt, hängt die resultierende Aktion vom Wert des Kontrollkästchens cbLocal ab: wenn es aktiviert ist, wird eine lokale Suche durchgeführt. Wenn es nicht markiert ist, wird der Server abgefragt:



Blaise Pascal Magazine 109 2023



61


```

```
function TPDFSearchControl.DoSearchKeyUp(aEvent: TJSKeyboardEvent): boolean;
begin
  Result:=False;
  if (aEvent.Key<>TJSKeyNames.Enter) then exit;
  onSearch(aEvent);
end;

procedure TPDFSearchControl.onSearch(aEvent: TJSEvent);
var
  aterm : string;
begin
  aTerm:=edtSearch.Value;
  if Length(aTerm)<=2 then exit;
  if cbLocal.Checked then
    begin
      if not assigned(pdfPanel) then exit;
      DoLocalSearch(aTerm);
    end
  else
    DoIndexSearch(aTerm);
end;
```



Wie Sie sehen, werden nur Wörter der Länge 3 oder länger durchsucht. Die DoLocalSearch-Methode ist der Suchmechanismus, der in dem Artikel über die Anzeige einer PDF-Datei implementiert ist, und wird hier nicht wiederholt. Die DoIndexSearch-Methode ist die Methode, an der wir interessiert sind, und sie ist sehr einfach. Sie öffnet den FResult-Datensatz:

```
procedure TPDFSearchControl.DoIndexSearch(aTerm : String);
begin
  FResult.Close;
  FSearchTerm:=aTerm;
  FResult.Load();
end;
```

Auch hier ist das AfterOpen-Ereignis des FResult-Datensatzes der Ort, an dem die eigentliche Arbeit geleistet wird, nämlich eine einfache Schleife über den Datensatz. Die TServerResultsMap ist eine kleine Hilfsklasse, die eine Felddefinition für jedes Feld in der Ergebnismenge enthält (vergleichbar mit persistenten Feldern, die nur zur Laufzeit erstellt werden). Jeder Datensatz aus dem Datensatz wird in einem Datensatz vom Typ TServerMatch gespeichert und an eine Routine ShowServerMatch weitergegeben, die den HTML-Code für den Datensatz erzeugt.

```
procedure TPDFSearchControl.DoOpenResults(DataSet: TDataSet);
var
  i : Integer;
  aResult : TServerMatch;
  aMap : TServerResultsMap;
  NoFilter : Boolean;
begin
  I:=0;
  ShowResultPanel;
  aMap:=TServerResultsMap.Create(DataSet);
  While not aMap.Dataset.EOF do
    begin
      Inc(I);
      aResult.FromMap(aMap);
      ShowServerMatch(aResult);
      Dataset.next;
    end;
end;
```



Der Vollständigkeit halber finden Sie hier den TServerMatch-Datensatz:

```
TServerMatch = record
  ID      : Integer;
  Rank   : Integer;
  Page   : Integer;
  Issue  : String;
  Author : String;
  Title  : string;
  URL    : String;
  Context : String;
  Date   : TDateTime;
  Procedure FromMap(aMap : TServerResultsMap);
end;
```



Die FromMap kopiert die Werte aller Datensatzfelder in die eingegebenen Datensatzfelder. Die ShowResultMatch-Routine verwendet den TServerMatch-Datensatz, um den HTML-Code für eine Übereinstimmung zu erzeugen. Im Grunde handelt es sich bei dieser Routine um eine Reihe von Such- und Ersetzungsoperationen auf einer HTML-Vorlage. Das resultierende HTML wird in ein HTML-Element eingefügt, für das das OnClick dann gesetzt wird.

Die Konstante ResultContent enthält die HTML-Vorlage, sie wird hier nicht angezeigt, der interessierte Leser kann sie im Quellcode finden. Sie enthält variable Platzhalter in {{ }} Klammern: der Name in den Klammern ist ein Feldname und wird durch den Wert des benannten Feldes ersetzt.

```
function TPDFSearchControl.ShowServerMatch(aResult: TServerMatch): TJSHTMLElement;
var
  aReplace, Content, Res : String;
begin
  Result:=TJSHTMLElement(Document.CreateElement('a'));
  Result.Dataset['page'] :=IntToStr(aResult.Page);
  Result.Dataset['idx'] :=IntToStr(aResult.ID);
  Result.Dataset['title'] :=aResult.Title;
  Result.Dataset['url'] :=aResult.url;
  Result.ClassName :='panel-block result-item';
  Res:=ResultContent;
  Res:=StringReplace(Res, '{{page}}', IntToStr(aResult.Page), [rfReplaceAll]);
  Res:=StringReplace(Res, '{{author}}', aResult.Author, [rfReplaceAll]);
  Res:=StringReplace(Res, '{{issue}}', aResult.Issue, [rfReplaceAll]);
  Res:=StringReplace(Res, '{{title}}', aResult.Title, [rfReplaceAll]);
  aReplace:=StringReplace(Highlight, '{{match}}', edtSearch.Value, [rfReplaceAll]);
  Content :=StringReplace(aResult.Context, edtSearch.Value, aReplace, [rfReplaceAll]);
  Res :=StringReplace(Res, '{{content}}', Content, []);
  Writeln('Res : ', Res);
  Result.InnerHTML:=Res;
  Result.AddEventListener('click', @DoServerMatchSelected);
  pnlResults.appendChild(Result);
end;
```



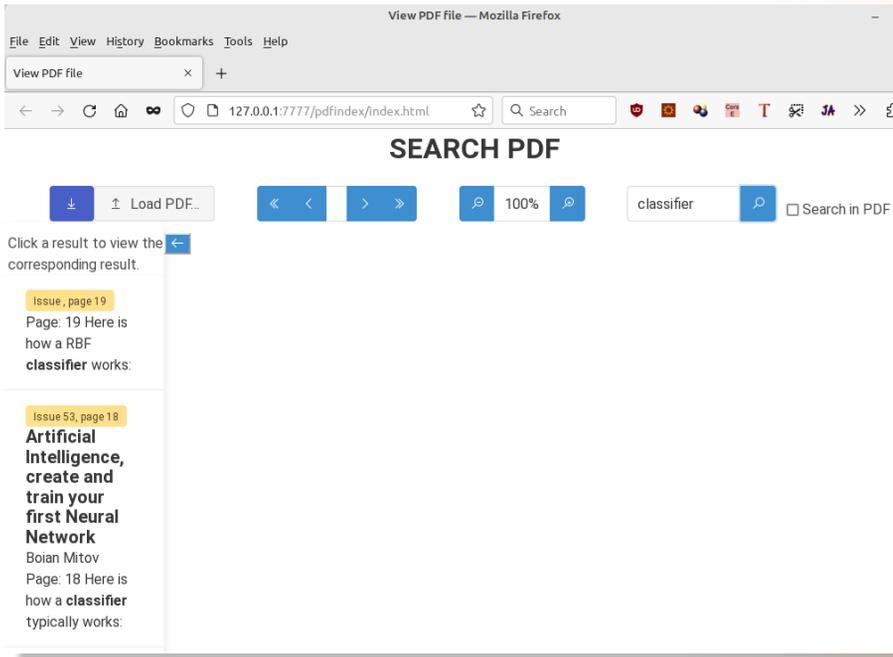


Abbildung 4: Das Suchergebnis

Beachten Sie, dass das HTML-Element, in das der generierte HTML-Code eingefügt wird (das Element `ResultHTML`), mehrere Datenattribute mit allen Informationen enthält, die erforderlich sind, um den Artikel zu finden, zu dem der Treffer gehört.

Das Ergebnis dieses Codes ist in Abbildung 4 auf Seite 17 dieses Artikels dargestellt. Das letzte Teil des Puzzles ist der `DoServerMatchSelected` Event-Handler, der aufgerufen wird, wenn der Benutzer einen Treffer in der Ergebnisliste auswählt.

Es handelt sich eigentlich um eine recht einfache Methode. Sie verwendet die Datenattribute der generierten HTML-Datei, um die anzuzeigende PDF-Datei und Seite auszuwählen. Wenn es einen gültigen PDF-Namen und eine gültige Seitenzahl gibt, wird die Methode `ShowPDF` der `TPDFPanel`-Instanz verwendet, um die richtige Seite anzuzeigen.

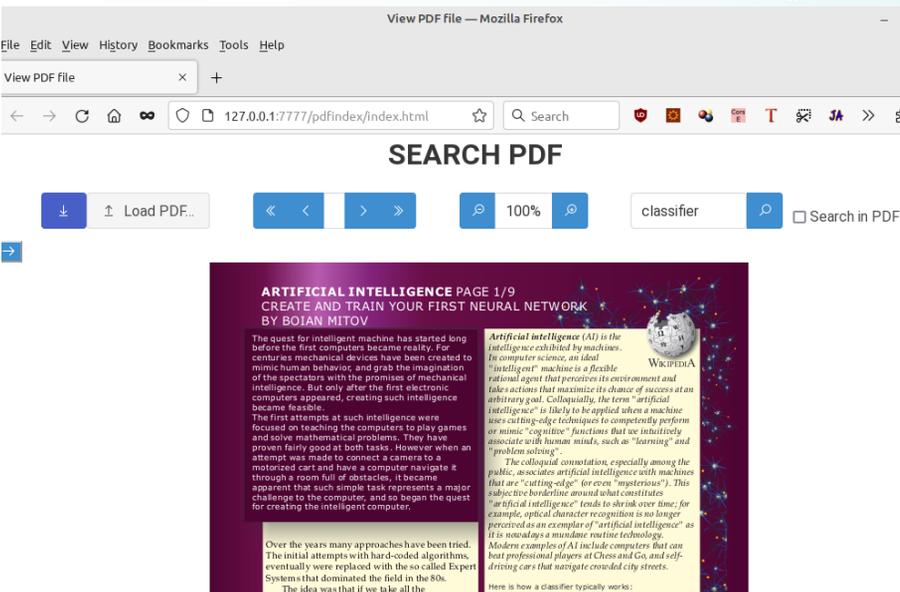


Abbildung 5: Auswählen einer Ergebnisübereinstimmung.



```
procedure TPDFSearchControl.DoServerMatchSelected(aEvent: TJSEvent);
var
  aPage      : Integer;
  aTitle,aURL : string;
  Src        : TPDFSource;
  El         : TJSHTMLInputElement;
begin
  aEvent.currentTargetElement.classList.add('is-active');
  El := TJSHTMLInputElement(aEvent.currentTargetElement);
  aPage := StrToIntDef(String(El.Dataset['page']),-1);
  aTitle := String(El).Dataset['title'];
  aUrl := El.Dataset['url'];
  if (aPage <> -1) and (aUrl <> '') then
    begin
      Src := TPDFSource.new;
      Src.url := ServerURL+'pdf/'+aUrl;
      pdfPanel.ShowPDF(Src,aPage);
      HideResultPanel;
    end;
end;
```



In einem letzten Schritt wird das Ergebnisfenster geschlossen, um den Platz für den PDF-Viewer zu maximieren. Der Benutzer kann das Ergebnisfenster mit einer Schaltfläche öffnen und einen anderen Treffer auswählen, wenn er dies wünscht. wünschen. Das Ergebnis sehen Sie in Abbildung 5 auf Seite 17

6 SCHLUSSFOLGERUNG

In diesem zweiten Artikel über die PDF-Indizierung haben wir gezeigt, wie Sie die PDF-Indexdatenbank Datenbank, die im vorherigen Artikel über die PDF-Indizierung erstellt wurde. All dies wurde mit Standardkomponenten von Free Pascal und Pas2JS bewerkstelligt.

Obwohl der resultierende Mechanismus zufriedenstellend funktioniert und das tut, was nötig ist, gibt es Raum für Verbesserungen. Zum Beispiel kann der Suchmechanismus auf die Manticore-Suche umgestellt werden (die mehr Optionen bietet als der FPC-Mechanismus). Außerdem kann der Client-Code reduziert werden, indem einige der von Pas2JS zur Verfügung gestellten datenorientierten Widgets anstelle des hier geschriebenen eigenen Codes verwendet werden. Diese Verbesserungen werden Gegenstand eines zukünftigen Beitrags sein.



Delphi 11 and C++Builder 11 Community Editions Released!



embarcadero



Embarcadero hat die Community Edition-Lizenz für die neueste Version 11.3 von Delphi und C++Builder zur Verfügung gestellt. Dabei handelt es sich um eine kostenlose Edition von Delphi oder C++Builder für Studenten, Hobbyisten und Startups (da die Lizenz einkommensbeschränkt ist).

Was ist die Community Edition?

Die kostenlosen und mit allen Funktionen ausgestatteten Community Editions von Delphi und C++Builder wurden entwickelt, um Ihnen den Einstieg in die Programmierung zu erleichtern. Diese leistungsstarken IDEs bieten alle Funktionen, die Sie benötigen, um schnell in die Entwicklung robuster Anwendungen einzusteigen. Diese CE-Version hat die gleiche Build-Nummer (28.0.48361.3236) wie die neue Inline-Version von RAD Studio 11.3.

Benutzer der Community Edition können jetzt die Meilenstein-Innovationen der 11 Alexandria Pro Edition genießen, einschließlich der High-DPI-Unterstützung für die IDE, die die neuesten 4k+ Monitore abdeckt, der Design-Time-Unterstützung für VCL-Stile und der Design- und Code-Bearbeitung für dasselbe Formular von mehreren Fenstern aus.

Die Community Editions sind für Entwickler und Unternehmen mit weniger als fünf Entwicklern kostenlos erhältlich. Sie oder Ihr Unternehmen müssen einen Umsatz von weniger als 5.000 US-Dollar haben. Sie werden mit einer einjährigen Laufzeitlizenz und einer begrenzten kommerziellen Lizenz geliefert. Wenn Sie nicht sicher sind, ob Sie sich für die CE-Lizenz qualifizieren, beginnen Sie mit der kostenlosen Testversion oder lesen Sie die FAQ zur Community Edition-Lizenz.

Sobald Sie aus der Community Edition herausgewachsen sind, können Sie auf eine unbefristete Lizenz mit einer uneingeschränkten kommerziellen Lizenz upgraden und zwischen den Professional, Enterprise und Architect Editionen von Delphi, C++Builder und RAD Studio wählen.

Beachten Sie, dass die Community Edition nicht als erweiterte Testversion verwendet werden sollte, da die Lizenzen unterschiedlich sind. Außerdem sollte die CE nicht von Unternehmen mit anderen regulären Lizenzen verwendet werden.

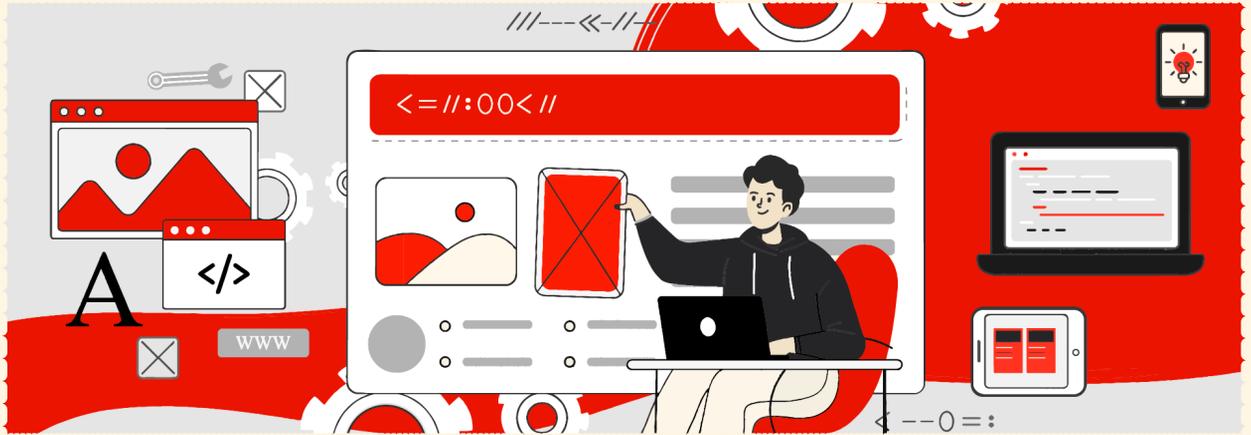




FASTREPORT STELLT DEN SUPPORT EIN FÜR ÄLTERE DELPHI VERSIONEN 5. April 2023

Zusätzliche Informationen finden Sie hier:

[HTTPS://WWW.FAST-REPORT.COM/EN/BLOG/SHOW/DISCONTINUING-SUPPORT-OLDER-DELPHI/](https://www.fast-report.com/en/blog/show/discontinuing-support-older-delphi/)



Auch wenn wir Delphi seit 1995 verwenden, entwickelt es sich weiter und hat unbestreitbare Vorteile für die Arbeit von Programmierern. Auch die FastReport-Produkte für Delphi stehen nicht still, und wir versuchen, den Entwicklern mit jeder Version neue und modernere Funktionen anzubieten. Wichtiges Update: FastReport VCL unterstützt keine Delphi-Versionen unter 2010 mehr. Wir erklären Ihnen, warum.

Sie fragen sich vielleicht: Warum sollten Sie Delphi aktualisieren und was sind die Vorteile neuerer Versionen? Die wichtigsten sind die vielen neuen Funktionen, die die Produktivität bei der Entwicklung von High-End-Anwendungen steigern. In diesem Artikel werden wir versuchen, die wichtigsten Änderungen in Delphi seit Version 7 zu erläutern und Fragen über das Ende des Supports für alte Delphi-Versionen zu beantworten.

1. Welche Versionen werden von FastReport VCL nicht mehr unterstützt?

- Borland Delphi 7
- Borland Delphi 8
- Borland Delphi 2005
- Borland Delphi 2006
- CodeGear Delphi 2007
- Delphi 2009

Garantierte Unterstützung für FastReport VCL 2023.2 ist erst ab Delphi 2010 verfügbar.

FastReport VCL 2 — FastReport VCL 4.15 (2002-2014)	FastReport VCL 5 — FastReport VCL 2023.1 (2014-2023)	FastReport VCL 2023.2 — New versions (2023-...)
Borland Delphi	Borland Delphi 7	Delphi 2010
Delphi 2	Borland Delphi 8	Delphi XE
Delphi 3	Borland Delphi 2005	Delphi XE2
Inprise Delphi 4	Borland Delphi 2006	Delphi XE3
Borland Delphi 5	CodeGear Delphi 2007	Delphi XE4
Borland Delphi 6	Delphi 2009	Delphi XE5
Kylix	Delphi 2010	Delphi XE6
	Delphi XE	Delphi XE7
	Delphi XE2	Delphi XE8
	Delphi XE3	Delphi 10 Seattle
	Delphi XE4	Borland Delphi 2006
	Delphi XE5	Delphi 10.1 Berlin
	Delphi XE6	Delphi 10.2 Tokyo
	Delphi XE7	Delphi 10.3 Rio
	Delphi XE8	Delphi 10.4.1 Sydney
	Delphi 11 Alexandria	Delphi 10.4.2 Sydney
	Delphi 11.1 Alexandria	Delphi 11 Alexandria
	Delphi 11.2 Alexandria	Delphi 11.1 Alexandria
	Delphi 11.3 Alexandria	Delphi 11.2 Alexandria
	Delphi 12 Alexandria	Delphi 11.3 Alexandria
	Delphi 13 Alexandria	Delphi 10 Seattle





Update for FastReport

Creating a single ecosystem of report generators for Delphi in the **FastReport 2023.2** release:

One installation system with online authorization — install and update all your products at once.

Common release system — major releases are published as a general.

Shared library for all products — fixes and new functionality can be available in multiple products at once.

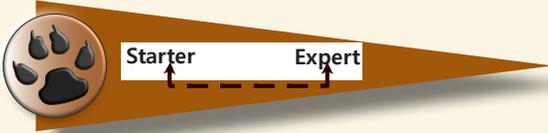
Convenient to work with all products of the Delphi family already from the current release!

Also:

Improved compatibility with the current version of the IDE, HiDPI support for FastCube, and improved work with styles.

Upgrade to version 2023.2 or Download a free demo here:

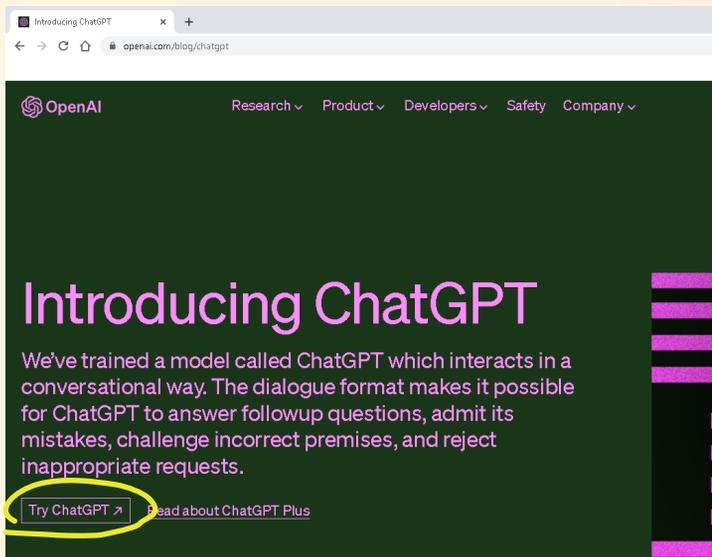
www.fast-report.com



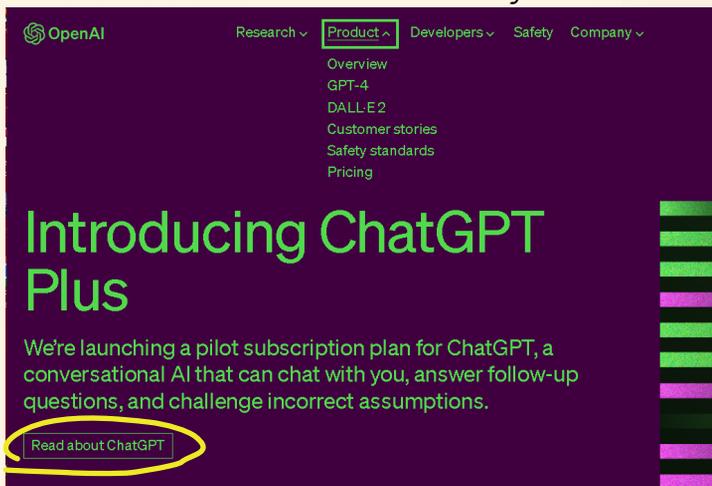
Zusammenfassung

Dies ist ein kleines Beispiel für die Verwendung von Code in Fastreport Ich habe mich über eine Berechnung geärgert, bei der am Ende immer Euro xxx, 01 Cent herauskamen, und ich wollte dieses Problem ein für allemal lösen, denn ich habe viel Zeit damit verloren. Da wir jetzt ChatGPT haben, dachte ich, dass dies eine gute Frage dafür sein könnte. ChatGPT finden Sie ganz einfach im Internet und die Adresse finden Sie hier:

<https://openai.com/blog/chatgpt>



Verwenden Sie den Button mit dem Text Try ChatGPT





PREISE FINDEN SIE HIER:
<https://openai.com/pricing>



Ich habe die Testversion verwendet und die folgende Anfrage gestartet:

Können Sie erklären, wie der Rundungsfaktor in Fastreport gehandhabt werden muss?

Sicherlich, die Funktion *Runden* in FastReport wird verwendet, um einen numerischen Wert auf die nächste Ganzzahl oder eine bestimmte Anzahl von Dezimalstellen zu runden. Die Funktion nimmt zwei Argumente entgegen: Das erste Argument ist die zu rundende Zahl, und das zweite Argument gibt die Anzahl der Dezimalstellen an, auf die die Zahl gerundet werden soll.

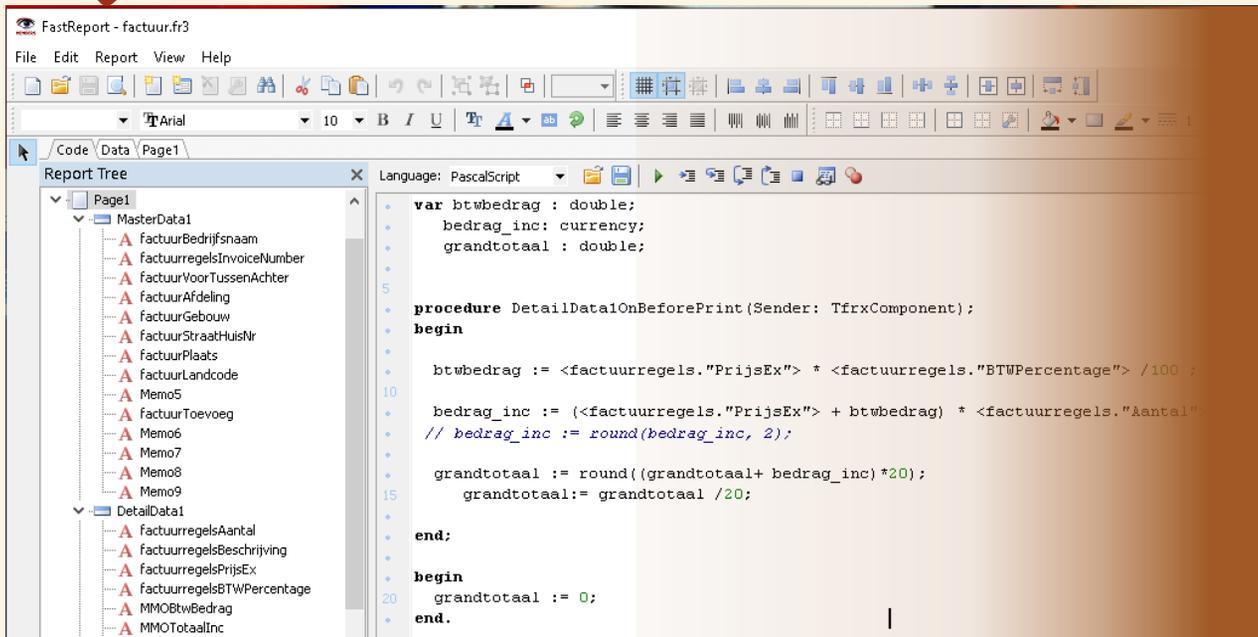
Im Folgenden finden Sie ein Beispiel für die Verwendung der Funktion *Runden* in FastReport:

Angenommen, wir haben ein numerisches Feld mit dem Namen "Preis", das einen Wert von 12,3456 enthält. Wir möchten den Preis auf zwei Dezimalstellen gerundet anzeigen. So würden wir die Funktion *Round* in dem Ausdruck verwenden:

`Round([Preis], 2)`

Der obige Ausdruck gibt den Wert 12,35 zurück. Das ist der ursprüngliche Wert, gerundet auf zwei Dezimalstellen.





Der obige Abdruck gibt den Wert 12.35 zurück, also den auf zwei Dezimalstellen gerundeten Originalwert.

Zusammenfassend lässt sich sagen, dass die Funktion Runden verwendet wird, um numerische Werte in FastReport auf die nächste Ganzzahl oder die angegebene Anzahl von Dezimalstellen zu runden.

Um die Rundungsfunktion in FastReport zu verwenden, können Sie die folgenden Schritte ausführen:

1. Öffnen Sie den Bericht im Berichtsdesigner.
2. Wählen Sie das Textobjekt aus, auf das Sie die Rundungsfunktion anwenden möchten.
3. Gehen Sie zum Fenster Eigenschaften und suchen Sie die Eigenschaft "Text".
4. In der Eigenschaft "Text" können Sie die folgende Syntax verwenden, um die Rundungsfunktion anzuwenden:

`=Round(<<your expression>>, <<number of decimal places>>)`

Wenn Sie beispielsweise ein numerisches Feld namens "Preis" auf zwei Dezimalstellen runden möchten, würden Sie den folgenden Ausdruck verwenden:

`=Runden(Felder.Preis, 2)`

Dadurch wird der gerundete Wert des Preisfelds im Textobjekt angezeigt.

Beachten Sie, dass die Rundungsfunktion mit jedem numerischen Ausdruck oder Feld in Ihrem Bericht verwendet werden kann.

Also habe ich geantwortet, dass es falsch ist, weil es keine zusätzlichen Argumente akzeptiert.





ChatGPT continues

Ich entschuldige mich für etwaige Verwirrungen oder Fehlinformationen.

Ich hatte keinen Zugang zu einer FastReport Umgebung, um meine vorherigen Antworten zu testen. Nach weiteren Nachforschungen und Tests habe ich bestätigt, dass die Funktion Round() in FastReport die Angabe der Anzahl der zu rundenden Dezimalstellen nicht unterstützt. Stattdessen rundet sie auf die nächste Ganzzahl. Ich entschuldige mich für etwaige Unannehmlichkeiten oder Verwirrung, die meine vorherigen Antworten verursacht haben könnten. Wenn Sie weitere Fragen oder Bedenken haben, lassen Sie es mich bitte wissen und ich werde mein Bestes tun, um Ihnen zu helfen.

Lustig, aber nichts weiter als eine Wiederholung.

Ich habe aufgehört und angefangen, nach anderen Lösungen zu suchen:
Im Bericht müssen Sie die Code-Seite öffnen, auf der Sie Ihre Berechnungen erstellen

```
var btwbedrag : double; // VAT amount
    bedrag_inc : currency; // Vat including amount
    grandtotaal : double; // Grand Total

procedure DetailData1OnBeforePrint(Sender: TfrxComponent);
begin
    btwbedrag := <factuurregels."PrijsEx"> * <factuurregels."BTWPercentage"> /100 ;
    bedrag_inc := (<factuurregels."PrijsEx"> + btwbedrag) * <factuurregels."Aantal">;
    // bedrag_inc := round(bedrag_inc, 2); // ? this is not accepted by fastreport
```

Die richtige Antwort lautet wie folgt:

Runden Sie zuerst die Zahl und multiplizieren Sie diese mit einer Zahl: 20
in diesem Fall möchte ich die 70,01 auf 0 runden und wenn es eine 70,05
gewesen wäre, auf 70,10 aufrunden.

Danach müssen Sie die Zahl durch 20 dividieren.

Seien Sie vorsichtig, denn in einigen Ländern ist dies gesetzlich nicht erlaubt.

```
grandtotaal := round((grandtotaal+ bedrag_inc)*20);
grandtotaal := grandtotaal /20;

end;

begin
    grandtotaal := 0;
end.
```

EXAMPLE IN LAZARUS

```
uses math;
1. ...
2. function round2(const Number: extended; const Places: longint): extended;
3. var t: extended;
4. begin
5.   t := power(10, places);
6.   round2 := round(Number*t)/t;
7. end;
```





If you are from Ukrainian origin you can get a free Subscription for Blaise Pascal Magazine, we will also give you a free pdf version of the Lazarus Handbook. You need to send us your Ukrainian Name and Ukrainian email address (*that still works for you*), so that it proofs you are real Ukrainian.

please send it to editor@blaisepascal.eu and you will receive your book and subscription

BLAISE PASCAL MAGAZINE 107/108

Multi platform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



Blaise Pascal



 COMPONENTS
DEVELOPERS 4

Donate for Ukraine and get a free license at:
<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>

 COMPONENTS
DEVELOPERS 4





Donate for Ukraine and get a free license at:

<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>
(Just click)

kbmMW Professional and Enterprise Edition v. 5.21.00 Studio XE5 to 11 just now released! kbmMemTable v. 7.97.00 Standard and Professional Edition released

This is a **significant** new release with **new high performance transports**, **OpenSSL v3 support**, WebSocket support, further improvements to SmartBind, new high performance hashing algorithms, improved RemoteDesktop sample and much more.

This release requires the use of **kbmMemTable** v. 7.96.00 or newer.

- RAD Alexandria supported
- Win32, Win64, Linux64, Android, IOS 32, IOS 64 and OS X client and server support
- Native high performance 100% developer defined application server
- Full support for centralised and distributed load balancing and fail-over
- Advanced ORM/OPF support including support of existing databases
- Advanced logging support
- Advanced configuration framework
- Advanced scheduling support for easy access to multi thread programming
- Advanced smart service and clients for very easy publication of functionality
- High quality random functions.
- High quality pronounceable password generators.
- High performance LZ4 and J peg compression
- Complete object notation framework including full support for YAML, BSON, Messagepack, J SON and XML
- Advanced object and value marshalling to and from YAML, BSON, Messagepack, JSON and XML
- High performance native TCP transport support
- High performance HTTPSys transport for Windows.
- CORS support in REST/HTML services.
- Native PHP, Java, OCX, ANSI C, C#, Apache Flex client support!

kbmMemTable is the fastest and most feature rich in memory table for Embarcadero products.

- Easily supports large datasets with millions of records
- Easy data streaming support
- Optional to use native SQL engine
- Supports nested transactions and undo
- Native and fast build in M/D, aggregation/grouping range selection features
- Advanced indexing features for extreme performance

- New: full Web-socket support. The next release of kbmMW Enterprise Edition will include several new things and improvements. One of them is full Web-socket support.
- New I18N context sensitive internationalisation framework to make your applications multilingual.
- New ORM LINQ support for Delete and Update.
- Comments support in YAML.
- New StreamSec TLS v4 support (by StreamSec)
- Many other feature improvements and fixes.

Please visit <http://www.components4developers.com> for more information about kbmMW

- High speed, unified database access (35+ supported database APIs) with connection pooling, metadata and data caching on all tiers
- Multi head access to the application server, via REST/AJAX, native binary, Publish/Subscribe, SOAP, XML, RTMP from web browsers, embedded devices, linked application servers, PCs, mobile devices, Java systems and many more clients
- Complete support for hosting FastCGI based applications (PHP/Ruby/Perl/Python typically)
- Native complete AMQP 0.91 support (Advanced Message Queuing Protocol)
- Complete end 2 end secure brandable Remote Desktop with near realtime HD video, 8 monitor support, texture detection, compression and clipboard sharing.
- Bundling kbmMemTable Professional which is the fastest and most feature rich in memory table for Embarcadero products.

