C-Entwicklung

Generated by Doxygen 1.10.0

Chapter 1

Projektname

Eine kurze Beschreibung Ihres Projekts. Dies sollte das Hauptziel und den Zweck Ihres Projekts klar umreißen.

1.1 Einleitung

Geben Sie eine detailliertere Beschreibung Ihres Projekts. Erklären Sie, was dieses Projekt macht und warum es nützlich ist.

1.2 Installation

Beschreiben Sie die Schritte zur Installation Ihres Projekts.

```
git clone https://github.com/ju1-eu/hello-world.git
cd hello-world
```

1.3 Verwendung

Zeigen Sie, wie das Projekt verwendet wird. Erklären Sie die verschiedenen Funktionen und wie sie genutzt werden können.

```
# Entwickeln auf Feature-Branches
# Dokumentationsupdates auf docs/update-readme
# Erstellen eines Branches für Dokumentationsupdates:
git branch -a
git log --oneline --graph --all
git checkout dev
git checkout -b docs/update-readme

# Durchführen von Änderungen an der Dokumentation:
vim README.md
vim .gitignore
git add .
git commit -m "Update README und .gitignore mit neuen Informationen"
git status

# Pushen des Dokumentationsbranches:
git push --set-upstream origin docs/update-readme

# Erstellen eines Pull Requests von `dev` nach `main`
# Mergen des Dokumentationsbranches in `dev`:
git checkout dev
git pull origin main
git merge docs/update-readme
```

```
git push origin dev

# Erstellen des Pull Requests von `dev` nach `main`:
gh pr create --base main --head dev --title "Aktualisierung der Dokumentation" --body "Fügt detaillierte
      Informationen zur README hinzu."

# Merge des Pull Requests:
# Liste aller offenen Pull Requests anzeigen [PR-Nummer]
gh pr list
gh pr view 2
#gh pr view 2 --web
# Code-Review durchführen
# Genehmigen eines Pull Requests
gh pr review 2 --approve
# PR ablehnen
#gh pr review 2 --request-changes "Bitte Code überprüfen"
gh pr merge 2
gh pr view 2
# Löschen des Feature-Branches:
git branch -d docs/update-readme
git push origin --delete docs/update-readme
git branch -a
git log --oneline --graph --all

# Regelmäßige Updates und Synchronisation
git checkout main
git pull origin main
git checkout dev
git push origin dev
git merge main
git branch -a
git log --oneline --graph --all
```

1.4 Mitwirken

Pull Requests sind willkommen. Für größere Änderungen öffnen Sie bitte zuerst ein Issue, um zu besprechen, was Sie ändern möchten.

1.5 Lizenz

MIT

1.6 Kontakt

Jan Unger – Website – esel573@gmail.com

Projektlink: https://github.com/ju1-eu/hello-world

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

File Documentation

3.1 datentypen_genauigkeit_berechnen.c File Reference

Demonstration der Genauigkeit verschiedener Fließkommadatentypen.

```
#include <stdio.h>
```

Functions

- int main (void)
  Hauptfunktion des Programms.

3.1.1 Detailed Description

Demonstration der Genauigkeit verschiedener Fließkommadatentypen.

Dieses Programm zeigt, wie die Präzision der Berechnung von Fließkommazahlen mit verschiedenen Datentypen variiert: float, double und long double.

3.1.2 Function Documentation

3.1.2.1 main()

```
int main (
            void  )
```

Hauptfunktion des Programms.

Initialisiert Variablen mit Fließkommawerten und führt Divisionen durch, um die Präzision der Datentypen float, double und long double zu demonstrieren.

Returns

Exit-Status des Programms.

< Float mit begrenzter Präzision

< Double mit höherer Präzision

< Long double mit potenziell höchster Präzision

< Ergebnis der Division als float

< Ergebnis der Division als double

< Ergebnis der Division als long double

3.2 datentypen_groessen.c File Reference

Größe (in Bytes) der verschiedenen Datentypen.

```
#include <stdio.h>
```

Functions

- int main (void)
  Hauptfunktion, zeigt die Größe verschiedener Datentypen in Bytes.

3.2.1 Detailed Description

Größe (in Bytes) der verschiedenen Datentypen.

3.2.2 Function Documentation

3.2.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, zeigt die Größe verschiedener Datentypen in Bytes.

Deklariert Variablen verschiedener Datentypen und einen Zeiger, initialisiert diese und gibt ihre Werte sowie ihre Größen in Bytes aus. Demonstrationszwecke für die Verwendung von sizeof.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

< Ganzzahlwert.

< Zeichenwert.

< Kurze Ganzzahl.

< Lange Ganzzahl.

< Sehr lange Ganzzahl.

< Fließkommazahl (einfache Genauigkeit).

< Fließkommazahl (doppelte Genauigkeit).

< Fließkommazahl (erweiterte Genauigkeit).

< Konstante Ganzzahl.

< Zeiger auf eine Ganzzahl.

3.3 datentypen_max_min.c File Reference

Zeigt die maximalen und minimalen Werte der verschiedenen Datentypen.

```
#include <float.h>
#include <limits.h>
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, zeigt die Maximal- und Minimalwerte verschiedener Datentypen.

3.3.1 Detailed Description

Zeigt die maximalen und minimalen Werte der verschiedenen Datentypen.

Dieses Programm demonstriert die Verwendung von Grenzwerten aus limits.h und float.h für verschiedene Datentypen in C, einschließlich ganzzahliger und Fließkommadatentypen.

3.3.2 Function Documentation

3.3.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, zeigt die Maximal- und Minimalwerte verschiedener Datentypen.

Nutzt Konstanten aus limits.h und float.h, um die Wertebereiche für verschiedene ganzzahlige und Fließkommadatentypen auszugeben. Dient der Illustration von Wertebereichen in der C Programmierung.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

3.4 datentypen_ueberlauf.c File Reference

Demonstration des Überlaufs bei einem unsigned short Datentyp.

```
#include <limits.h>
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, demonstriert den Überlauf bei einem unsigned short.

3.4.1 Detailed Description

Demonstration des Überlaufs bei einem unsigned short Datentyp.

Dieses Programm zeigt, was passiert, wenn ein unsigned short Datentyp seinen maximalen Wert überschreitet und wieder bei 0 anfängt. Es setzt den Wert nahe am maximalen Wert, erhöht ihn in einer Schleife und zeigt den Überlauf.

3.4.2 Function Documentation

3.4.2.1 main()

```
int main (
             void  )
```

Hauptfunktion, demonstriert den Überlauf bei einem unsigned short.

Beginnt mit einem Wert nahe dem maximalen Wert für unsigned short und erhöht diesen in einer Schleife mehrmals, um zu zeigen, wie der Wert nach Erreichen des maximalen Wertes auf 0 zurückgesetzt wird und von vorne beginnt.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

3.5 explizite_konvertierung.c File Reference

Demonstration der expliziten Konvertierung von double zu int in C.

```
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, führt die Konvertierung durch und zeigt das Ergebnis.

3.5.1 Detailed Description

Demonstration der expliziten Konvertierung von double zu int in C.

Dieses Programm zeigt, wie ein Wert vom Typ double explizit in einen Wert vom Typ int konvertiert wird. Es illustriert den Prozess und das Ergebnis der Konvertierung, einschließlich des Verlusts von Nachkommastellen.

3.5.2 Function Documentation

3.5.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, führt die Konvertierung durch und zeigt das Ergebnis.

Deklariert eine Variable vom Typ double und weist ihr einen Wert zu. Führt dann eine explizite Konvertierung (Casting) dieses Wertes zu einem int durch und speichert das Ergebnis in einer int-Variablen. Schließlich werden der ursprüngliche double-Wert und der konvertierte int-Wert ausgegeben.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

3.6 hallo-welt.c File Reference

Ein einfaches Hallo Welt Programm in C.

```
#include <stdio.h>
```

Functions

- int main ()
  Hauptfunktion des Programms.

3.6.1 Detailed Description

Ein einfaches Hallo Welt Programm in C.

Dieses Programm gibt die Nachricht "Hallo Welt" auf der Standardausgabe aus. Es dient als einführendes Beispiel für die Ausgabe in C-Programmen.

3.6.2 Function Documentation

3.6.2.1 main()

```
int main (
            void  )
```

Hauptfunktion des Programms.

Gibt eine Grußnachricht auf der Standardausgabe aus und beendet sich dann mit einem erfolgreichen Exit-Code.

Returns

int Rückgabewert des Programms. 0 signalisiert ein erfolgreiches Beenden.

3.7 implizite_konvertierung.c File Reference

Demonstration der impliziten Konvertierung in C.

```
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, führt die Addition durch und zeigt die implizite Konvertierung.

3.7.1 Detailed Description

Demonstration der impliziten Konvertierung in C.

Dieses Programm demonstriert die implizite Konvertierung von int zu float während der Addition eines int- und eines float-Wertes. Es zeigt, wie C automatisch einen Datentyp in einen anderen konvertiert, wenn es für die Operation erforderlich ist, um Typinkompatibilitäten zu vermeiden.

3.7.2 Function Documentation

3.7.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, führt die Addition durch und zeigt die implizite Konvertierung.

Deklariert eine int- und eine float-Variable und addiert diese. Die int-Variable wird dabei implizit zu einem float konvertiert, um die Addition korrekt durchführen zu können. Anschließend werden der ursprüngliche int-Wert, der float-Wert und das Ergebnis der Addition ausgegeben.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

3.8 rechteck_berechnung_v1.c File Reference

Berechnet Umfang und Fläche eines Rechtecks basierend auf Benutzereingaben.

```
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, die den Umfang und Flächeninhalt eines Rechtecks berechnet.

3.8.1 Detailed Description

Berechnet Umfang und Fläche eines Rechtecks basierend auf Benutzereingaben.

Dieses Programm fordert den Benutzer auf, die Längen der Seiten eines Rechtecks einzugeben. Anschließend berechnet es den Umfang und den Flächeninhalt des Rechtecks und gibt diese Werte aus.

3.8.2 Function Documentation

3.8.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, die den Umfang und Flächeninhalt eines Rechtecks berechnet.

Fordert den Benutzer zur Eingabe der Längen von zwei Seiten eines Rechtecks auf. Berechnet dann den Umfang und Flächeninhalt des Rechtecks mit den Formeln: Umfang = 2 $*$ (seite_a + seite_b) Flächeninhalt = seite_a $*$ seite_b und gibt die berechneten Werte aus.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

$<$ Länge der Seite a des Rechtecks.

$<$ Länge der Seite b des Rechtecks.

$<$ Berechneter Umfang des Rechtecks.

$<$ Berechneter Flächeninhalt des Rechtecks.

3.9 rechteck_berechnung_v2.c File Reference

Berechnet Umfang und Fläche eines Rechtecks mit validierter Benutzereingabe.

```
#include <stdio.h>
```

Functions

- void bereinigeEingabePuffer ()
  Bereinigt den Eingabepuffer.
- int main (void)
  Hauptfunktion, die die Benutzereingaben entgegennimmt, validiert und die Berechnungen durchführt.

3.9.1 Detailed Description

Berechnet Umfang und Fläche eines Rechtecks mit validierter Benutzereingabe.

Dieses Programm fordert den Benutzer auf, die Längen der Seiten eines Rechtecks einzugeben, validiert die Eingaben als positive Dezimalzahlen und berechnet dann den Umfang und die Fläche des Rechtecks. Ungültige Eingaben führen zu einer erneuten Eingabeaufforderung.

3.9.2 Function Documentation

3.9.2.1 bereinigeEingabePuffer()

```
void bereinigeEingabePuffer ( )
```

Bereinigt den Eingabepuffer.

Diese Funktion entfernt alle übrig gebliebenen Zeichen aus dem Eingabepuffer bis zum nächsten Zeilenumbruch oder EOF. Sie wird verwendet, um den Puffer nach einer ungültigen Eingabe zu bereinigen, damit nachfolgende Leseversuche nicht fehlschlagen.

3.9.2.2 main()

```
int main (
            void  )
```

Hauptfunktion, die die Benutzereingaben entgegennimmt, validiert und die Berechnungen durchführt.

Fordert den Benutzer zur Eingabe der Längen von zwei Seiten eines Rechtecks auf und validiert diese Eingaben. Berechnet dann den Umfang und die Fläche des Rechtecks mit den Formeln: Umfang = 2 $*$ (seite_a + seite_b) Flächeninhalt = seite_a $*$ seite_b und gibt die berechneten Werte aus.

Returns

int Rückgabewert des Programms. 0 für erfolgreiches Beenden.

$<$ Länge der Seite A des Rechtecks.

$<$ Länge der Seite B des Rechtecks.

$<$ Berechneter Umfang des Rechtecks.

$<$ Berechneter Flächeninhalt des Rechtecks.

3.10 variablen_datentypen.c File Reference

Demonstration von Variablen, Datentypen, Deklaration, Initialisierung, Konstanten und Zeigern in C.

```
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, die die Deklaration, Initialisierung und Basisausgabe von Variablen demonstriert.

3.10.1 Detailed Description

Demonstration von Variablen, Datentypen, Deklaration, Initialisierung, Konstanten und Zeigern in C.

Dieses Programm deklariert und initialisiert Variablen unterschiedlicher Datentypen, eine Konstante und einen Zeiger. Anschließend werden die Werte und Adressen ausgegeben, um die grundlegende Syntax und Operationen in C zu demonstrieren.

3.10.2 Function Documentation

3.10.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, die die Deklaration, Initialisierung und Basisausgabe von Variablen demonstriert.

Deklariert Variablen verschiedener Datentypen, eine Konstante und einen Zeiger. Zeigt, wie Werte und Zeigeradressen ausgegeben werden. Demonstriert die Verwendung des `const`-Schlüsselworts und Zeigeroperationen.

Returns

int Rückgabestatus des Programms. Gibt 0 zurück, was eine erfolgreiche Ausführung anzeigt.

< Integer-Variable.

< Zeichen-Variable.

< Kurze Integer-Variable.

< Lange Integer-Variable.

< Sehr lange Integer-Variable.

< Fließkommazahl-Variable.

< Fließkommazahl-Variable mit doppelter Genauigkeit.

< Fließkommazahl-Variable mit erweiterter Genauigkeit.

< Konstante Integer-Variable.

< Zeiger auf eine Integer-Variable.

3.11 variablen_und_rechenoperationen.c File Reference

Demonstration der Verwendung von Variablen und grundlegenden arithmetischen Operationen in C.

```
#include <stdio.h>
```

Functions

- int main (void)

  Hauptfunktion, die arithmetische Operationen mit Variablen demonstriert.

3.11.1 Detailed Description

Demonstration der Verwendung von Variablen und grundlegenden arithmetischen Operationen in C.

Dieses Programm deklariert und initialisiert verschiedene Arten von Variablen und führt grundlegende arithmetische Operationen wie Addition, Multiplikation, Division (einschließlich Fließkommadivision), Inkrement, Dekrement und Modulo-Operation durch. Es zeigt, wie diese Operationen durchgeführt und die Ergebnisse ausgegeben werden.

3.11.2 Function Documentation

3.11.2.1 main()

```
int main (
            void  )
```

Hauptfunktion, die arithmetische Operationen mit Variablen demonstriert.

Führt verschiedene arithmetische Operationen durch, einschließlich Addition, Multiplikation mit Zuweisung, Fließkommadivision, Inkrement- und Dekrement-Operationen, sowie die Modulo-Operation und gibt die Ergebnisse aus. Demonstriert die Syntax für diese Operationen in C.

Returns

int Rückgabestatus des Programms. Gibt 0 zurück, was eine erfolgreiche Ausführung anzeigt.

< Zählervariable.

< Fließkommavariable für die durchschnittliche Temperatur.

< Integer-Variable für die Summe.

< Fließkommavariable für das Divisionsergebnis.

< Integer-Variable für das Produkt.

Index