**Taschenrechner**

**Von**

**Diego, Gino, Jevgenia, Lynn**

# IPERKA:

## Informieren:

***Auftrag:***

Einen Taschenrechner programmieren, der folgendes kann: Summieren, Subtrahieren, Dividieren, Multiplizieren, Exponent, Wurzeln, Fakultät + ein Wahlfeature.

Zusätzlich muss noch gemacht werden: IPERKA, Dokumentation, PPP, Präsentation, Informationsbeschaffung.

***Abgabe:***

26.09.24 um 1100

***Einteilung:***

-Summieren, Subtrahieren, Dividieren, Multiplizieren

-Exponent

-Wurzeln

-Fakultät

**Unser gewähltes Wahlfeature:**

-Schere, Stein, Papier

- BMI Rechner

**Gino;**

-Summieren, Subtrahieren, Dividieren, Multiplizieren

**Diego;**

-Exponent

-Schere Stein Papier

-BMI Rechner

**Jevgenia;**

-Wurzeln

**Lynn;**

-Fakultät

## Planung:

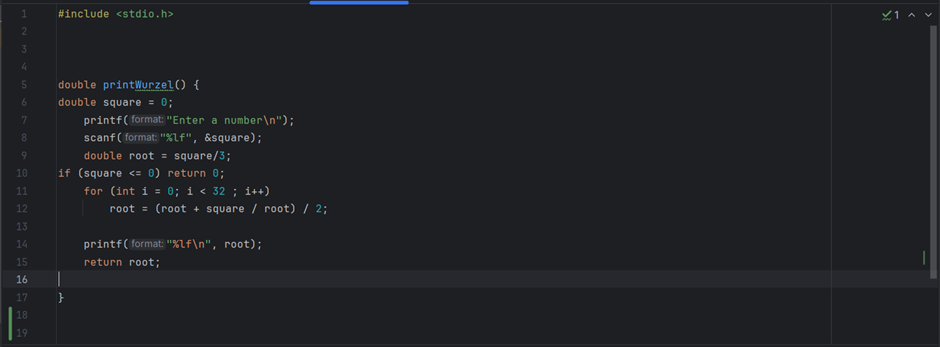
|  |  |
| --- | --- |
| **Zeitplan** |  |
| 19.09.24 | -Planung  -Aufteilung  -Anfangen zu programmieren |
| 20.09.24 | -weiter programmieren  -Main verbinden |
| 25.09.24 | -Programme einfügen ins Main  -Dokumentation schreiben  -Taschenrechner fertig |
| 26.09.24 | -Präsentation  -Überarbeitung  -Dokumentation fertig machen |

***Meilensteine:***

1. Auftrag aufteilen
2. Hauptfeatures programmieren
3. Wahlfeature programmieren
4. Dokumentation schreiben
5. Präsentation erstellen

## Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Jevgenia** | **Tages Ziel**  **Erreichtes** | **Details** | **Sonstiges** | **Nächster Schritt** |
| 19.9.24 | **Ziel:**  -Planung,  -Dokumentation  -“I, P” fertig,  **Erreicht:**  -“Informieren”,  -“Planung”,  -GitHub  -eingerichtet,  -Begonnen mit  -programmieren, | **Meine Aufgabe:**  -Meine Aufgabe für die ersten Tage ist es, eine Funktion zu programmieren, mit welcher man Wurzeln berechnen kann.  Dafür muss ich zuerst mich über das Thema informieren und herausfinden, wie ich so eine Funktion erstellen kann | -Ich hatte zuerst Probleme damit, die richtigen Informationen zu den Wurzeln zu finden  -Wir hatten zu Beginn Probleme mit GitHub, wofür wir auch viel Zeit beansprucht hatten | -Wurzeln Berechnen fertig programmieren |
| 20.9.24 | **Ziel:**  -Programm (Wurzeln) fertig programmieren und kontrolliert  **Erreicht:**  -Wurzel Funktion programmiert  -Dokumentation weiterbearbeiten | **Meine Aufgabe:**  -Ich muss noch ein wenig mehr Recherchen betreiben, da ich noch unsicher bin, wie ich die Funktion erstellen soll.  -Danach muss ich die Funktion programmieren und kontrollieren. | -Ich hatte ein paar Probleme mit der Funktion. Der Code funktionierte an manchen Stellen nicht, was mir ein wenig Zeit gekostet hat | -Dokumentation weiterbearbeiten |
| 25.9.24 | **Ziel:**  -Programm fertig bearbeiten  -Dokumentation schreiben  **Erreicht:**  -Programm korrigiert und überarbeitet  -Dokumentation geschrieben | **Meine Aufgaben:**  -Ich muss meinen Teil der Dokumentation fertig schreiben. In der Gruppe müssen wir den “Realisieren” Teil anschauen. | - Wir hatten ein paar Probleme mit der Main. Wenn man da Programm wiederholen wollte, wurde es als endlose Schleife wiedergegeben. Wir konnten das Problem jedoch lösen.  - | -Präsentation erstellen |
| 26.9.24 | **Ziel:**  -Präsentation erstellen  **Erreicht:**  -Doku fertig | **Meine Aufgaben:**  -Dokumentation fertig schreiben.  -Projekt abgeben | - | **Ende** |



Ich habe die Berechnung der Wurzel mit dem Newton Verfahren (root = (root + square / root) / 2) programmiert. Mit dieser Rechnung wird der root als Schätzung berechnet, wobei die Rechnung 32-mal

wiederholt wird. Mit jeder Wiederholung wird das Resultat immer genauer. Für die Wiederholung ist das for (int i = 0; i < 32 ; i++) zuständig.

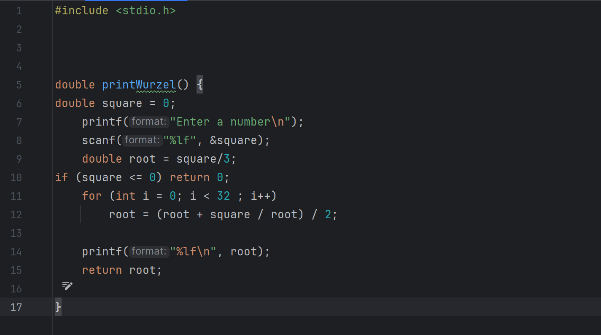
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Diego** | **Tages Ziel**  **Erreichtes** | **Details** | **Sonstiges** | **Nächster Schritt** |
| 19.9.24 | **Ziel:**  -IPERKA befolgen  -vielleicht mit TR anfangen  **Erreicht:**  -IPER Geschafft | -GIT erstellt, Projekt erstellt und begonnen zu programmieren | -Gab ein paar Probleme, aber die konnten gelöst werden | -An code weiterschreiben |
| 20.9.24 | **Ziel:**  -Taschenrechner zum Funktionieren bringen  **Erreicht:**  -DerTaschenrechner funktioniert und ich begann mit dem Zusatzfeature | -Ich habe die Exponenten zum funktionieren | -Ich hatte Probleme bei der Exponenten Funktion | -Den Taschenrechner erwteitern. |
| 25.9.24 | **Ziel:**  -Zusatz Funktion des Taschenrechners zum Funktionieren bringen  **Erreicht:**  -Zusatz Funktion des Taschenrechners zum laufen gebracht und weiter Funktionen eingebaut | -Ich wurde mit dem Schere, Stein, Papier fertig und ich habe noch eine BMI Rechner eingebaut | -Ich hatte kleine Probleme im Code diese konnte ich aber einfach lösen. | -Dokumentieren und/oder weitere Funktionen einbauen |
| 26.9.24 | **Ziel:**  -Taschenrechner fertigstellen, Dok. beenden  **Erreicht:**  -Taschenrechner fertigstellt, Dok. beendet | -Ich habe überprüft ob alles einwandfrei funktioniert danach habe ich an der Dokumentation gearbeitet | -Es gab keine Probleme | **Ende** |

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

1. Ich habe am Main gearbeitet und setzte den Taschenrechner zusammen
2. Ich habe den Code für die Exponente geschrieben
3. Ich habe den Schere, Stein, Papier code geschrieben
4. Ich habe den BMI rechenr code geschrieben

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lynn** | **Tages Ziel**  **Erreichtes** | **Details** | **Sonstiges** | **Nächster Schritt** |
| 19.9.24 | **Ziel:**  -Planen  -Dokumentation  -Einteilung  **Erreicht:**  -Planen  -Dokumentation  -Einteilung  -Programm beinahe fertig  -GitHub eingerichtet | -Die Dokumentation wurde vorbereitet und ausgefüllt.  **-**Ich habe das Programm angefangen zu schreiben, als ich nicht mehr weiterkam, fraget ich nach und es wurde mir erklärt. Daraufhin habe ich es verstanden und war auch so gut wie fertig nun muss es nur noch ins Main eingefügt werden. | -Die Einrichtung von GitHub war etwas verwirrend und kostete uns sehr viel Zeit.  -Gesavter CODE  int num();  int fakultaet1() {  int number;  int result;   result = num(number);   return result; }  int num(int number) {  int result = 1;  while (number != 0) {   result = result \* number;  number--;  }  return result; } | -Programm ins Main einfügen  -Dokumentation  -Programm ins GitHub pushen |
| 20.9.24 | **Ziel:**  -Dokumentation  -Programm ins Main  -Programm überarbeiten  **Erreichtes:**  -Programm fertig  -Dokumentation  -mit Main Verbinden | -Ich habe das Programm überprüft und fertig gestellt.  -Nach einer kurzen Besprechung begannen wir damit, alle Programme zusammen zu fügen. | -Wir waren erstaunlich schnell fertig mit unseren Teilen. Nu fehlt nur noch das Main und das Wahlfeature. | -Überarbeiten  -alles ins Main einfügen  -Dokumentation |
| 25.9.24 | **Ziel:**  -Programm fertig machen  -Dokumentation schreiben  **Erreicht:**  -Programm fertig und überarbeitet | -Wir haben alle Errors und Warnungen entfernt. Sonst wurde alles noch ins Englische übersetzt und benutzerfreundlicher gestaltet. | -Es gab keine Probleme mit GitHub und auch sonst wars sehr entspannt. | -PPP  -Präsentation |
| 26.9.24 | **Ziel:**  -PPP erstellen  -Präsentieren  -Dokumentation fertigstellen  **Erreicht:**  - | - | - | **Ende** |

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software enthält.

Automatisch generierte BeschreibungIch habe bei diesem Projekt die Fakultät programmiert. Zuerst musste ich herausfinden, wie man dies berechnet, da ich das nicht in der Sek. hatten. Verstanden hatte ich es schnell und begann daraufhin sofort zu coden. Da meine Funktion keinen Wert zurückgeben musste, wurde diese ein "void". Als erstes erstellte ich die Variable "number", dann baute ich ein "printf" ein, welches nach einer Nummer fragt. Mit dem "scanf" darunter wird der Wert von "number" neu festgelegt. Weil ich meinen Hauptteil möglichst kurzhalten wollte, erstellte ich unter halb eine weitere Funktion. "num" benannte ich diese. Da sie einen Wert zurückgeben musste, wurde dieser Teil des Codes ein "int", welcher mit einem "return result" endete. Damit CLion sich nicht beschwerte, fügte ich sofort die Variable "result" hinzu, gleichzeitig bestimmte ich ihren Wert auf "1". Nun kam der wichtigste Teil, die Berechnung der Fakultät. Eine "while" Schleife wurde erstellt, mit der bedingung "number != 0". Dies sorgt dafür, dass die Funktion beendet wird, sobald "number" den Wert "0" erreicht. In der Schleife wird der Wert von "result" neu festgelegt zu "result \* number". Danach wird von "number" eins abgezogen. Dies wiederholt sich so lange bis "number" gleich "0" ist. Dann wird der Wert von "result" in den Hauptteil eingefügt. Als nächstes erstellte ich eine dritte Variable, "result". Diese übernahm den zurückgegebenen Wert von "num" und darunter wird das Resultat gespeichert. Zum Schluss schrieb ich ganz oben noch eine Ankündigung von "num" hin.

Die Funktion war nun fertig und musste nur noch ins "main" eingefügt werden. Dazu erstellte ich das Headerfile "fakultaet.h". In diesem erwähnte ich den Hauptteil meiner Funktion und importierte es anschliessend mit "#include "fakultaet.h"" ins "main".

Wenn man nun im "main" Fakultät auswählt wird meine Funktion abgerufen. Diese fragt dann:

**Whats your number?**

Daraufhin kann man eine ganze Zahl eingeben. (**Zum Beispiel: 5**)

Dann wird "num" aufgerufen und überprüft ob unsere Zahl (**5**) gleich "0" ist. Zu diesem Zeitpunkt trifft dies nicht zu, also wird die "while" Schleife ausgeführt.

Nun wird result ausgerechnet.

**result(1\*5=5) = result(1) \* number(5)**

**number—(5-1=4)**

Es wird erneut kontrolliert ob "number" gleich "0" ist. Nein ist es nicht.

**result(5\*4=20) = result(5) \* number(4)**

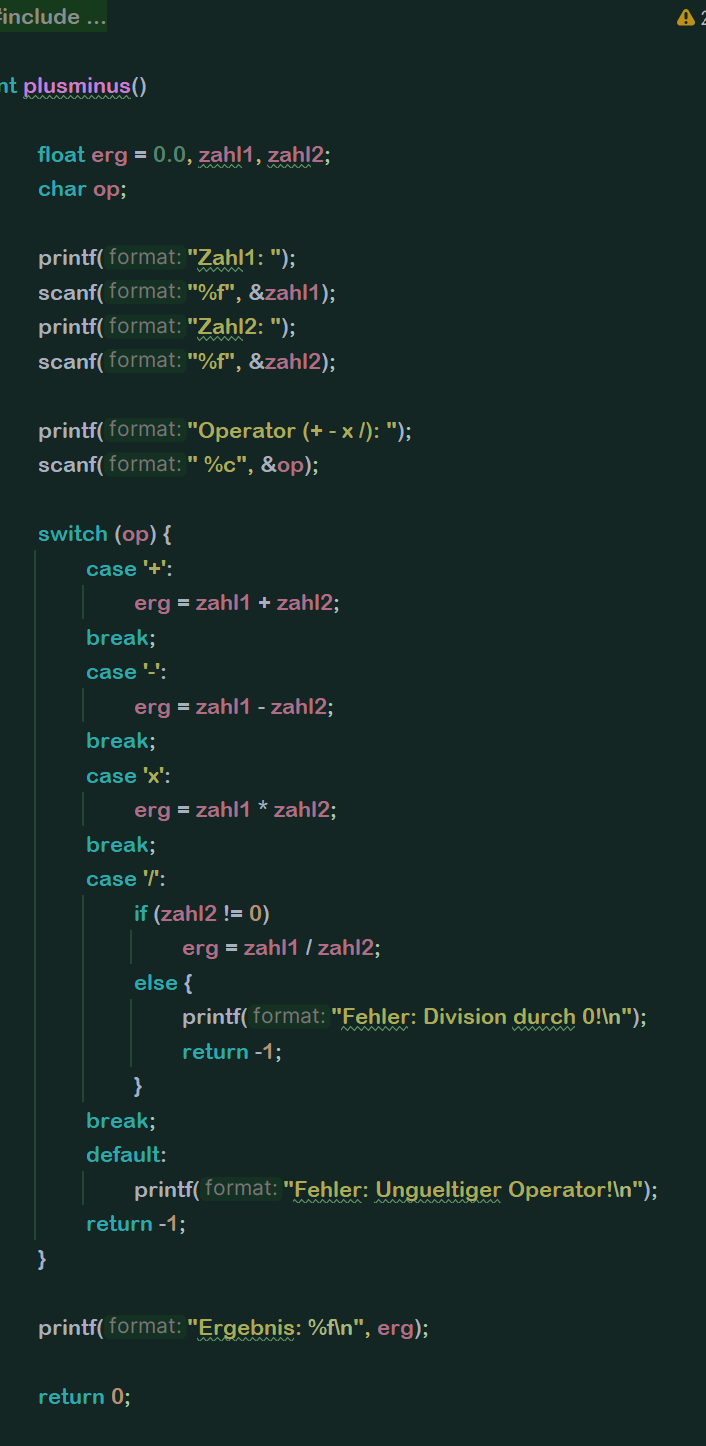
**number—(4-1=3)**

**->**

**result(1\*5=5) = result(1) \* number(5)**

**number—(5-1=4)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gino** | **Tages Ziel**  **Erreichtes** | **Details** | **Sonstiges** | **Nächster Schritt** |
| 19.9.24 | **Ziel:**  -Taschenrechner programmieren  **Erreicht:**  -Taschenrechner programmiert, jedoch noch nicht funktionierend | -Ich habe den Taschenrechner bereits programmiert aber, da er nicht korrekt mit dem Main etc. Verbunden wurde geht die Ausgabe noch nicht korrekt. | -Kurzer Unterbruch wegen dem fehlerhaften Gitignore | -Programm mit Main verknüpfen und den Taschenrechner überarbeiten |
| 20.9.24 | **Ziel:**  -Dokumentation  -Taschenrechner zum Funktionieren bringen  **Erreicht:**  -Dokumentation von erstem Tag | -Ich habe den Taschenrechner fertig programmiert | -Die Implementierung in das Main funktioniert noch nicht ganz | -In das Main implementieren |
| 25.9.25 | **Ziel:**  -In Main implementieren  **Erreicht:**  -In Main hinzugefügt | -Es funktioniert jetzt mit den anderen zusammen | -Das Problem ist mir nicht bekannt aber es funktioniert jetzt | -PPT anfangen |
| 26.9.24 | **Ziel:**  -PPT erstellen  **Erreicht:**  -PPT erstellt | -PPT zur präsentation erstellt und den anderen freigegeben | - | **Ende** |



Ich gebe zwei Zahlen und einen Operator (+, -, x, /) ein.

Das Programm führt die Rechenoperation für mich aus.

Ich sehe das Ergebnis oder eine Fehlermeldung.

## Entscheiden:

Gab es Entscheidungen? Wenn ja, welche? Wie haben wir sie getroffen?

Unsere erste Entscheidung war, wer welchen Teil der Hauptfeatures programmiert. Jeder hat gesagt, welchen Teil er/sie programmieren will und so konnten wir sofort beginnen. Als nächstes haben wir einen Zeitplan erstellt, wobei wir uns unsere Aufgaben eingeteilten. Danach mussten wir uns entscheiden welches Wahlfeature wir noch hinzufügen wollten und wer dieses programmieren soll. Diego erklärte sich dazu bereit dies zu übernommen. Nach dem wir das Programm beendet hatten, mussten wir uns entscheiden, wer welchen Teil der Dokumentation schreiben und die Präsentation vorbereiten würde. Wir haben uns schlussendlich dazu entschieden die Dokumentation zu beenden und die Präsentation direkt über CLion zu zeigen.

Welche Ressourcen werden benutzt?

* Unsere Laptops
* Internet für Recherchen

Welche Ressourcen stehen zur Verfügung?

* Laptops
* Internetverbindung
* C Vorwissen

Zusätzliche Ideen? Wenn Ja, welche?

* Intro “CALCULATOR”
* 2. Wahlfeature

## Kontrollieren

Nach dem wir das Projekt fertig programmiert hatten, mussten wir es auf Warnungen und Bugs überprüfen. Lynn und Jevgenia haben es nach dem Dokument “Definition of Done” auf diese kontrolliert.   
Unseren Zeitplan haben wir gut beachtet, sodass wir gegen Ende keinen Stress hatten. Wir haben unsere Meilensteine abgeschlossen und konnten alles in Zeit fertigstellen. Zuletzt haben wir die Dokumentation auch Rechtschreibefehler kontrolliert und bearbeitet.

## Auswerten:

Wir haben bei der Bearbeitung des Projektes bemerkt, dass wir uns nicht immer sicher waren, ob das Gemachte stimmte. Zum Beispiel, beim GitHub hatten wir erst im Nachhinein gemerkt, dass es besser gewesen wäre, wenn wir zuerst jemanden um Hilfe gebeten hätten, da wir ziemlich viel Zeit mit der Einrichtung verschwendeten.

Wir sind sehr zufrieden mit unserem Ergebnis. Der Taschenrechner erfüllt seinen Zweck. Die Meilensteine konnten wir alle abschliessen, genau wie die Dokumentation.

Was wir beim nächsten Mal besser machen könnten, wäre es, uns öfter in der Gruppe zu treffen und unsere Vorschritte festzuhalten.