



---

# RECHENTRAINER DOKUMENTATION

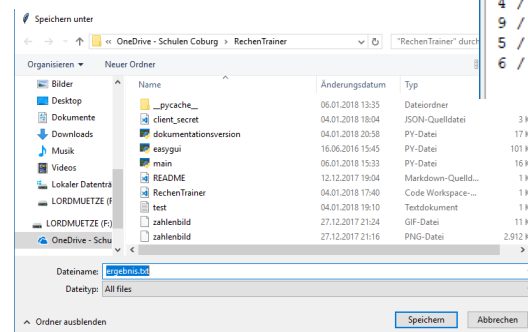
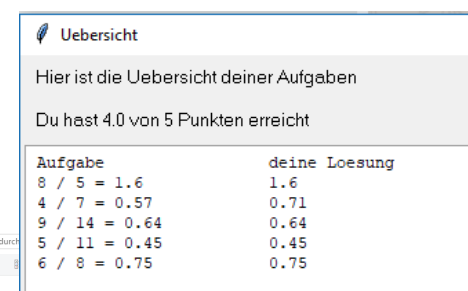
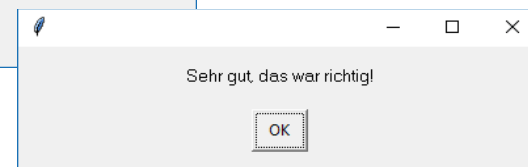
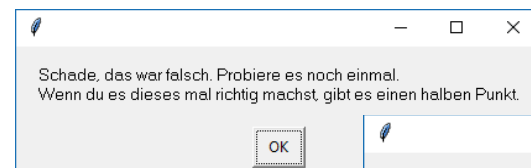
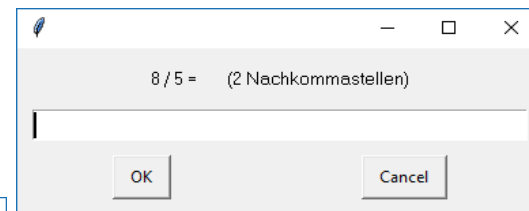
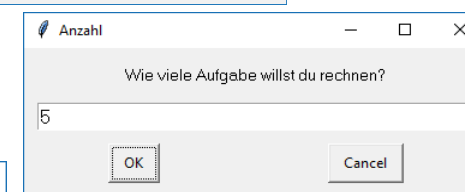
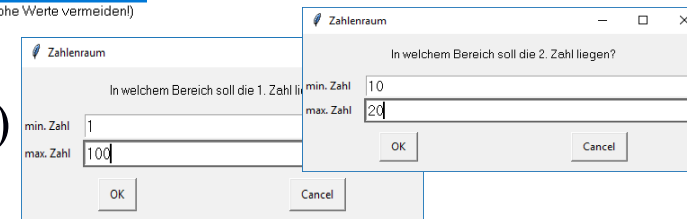
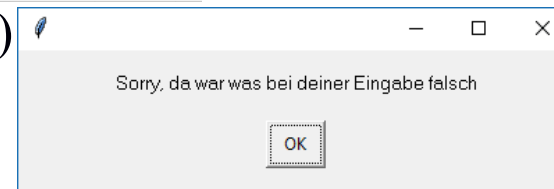
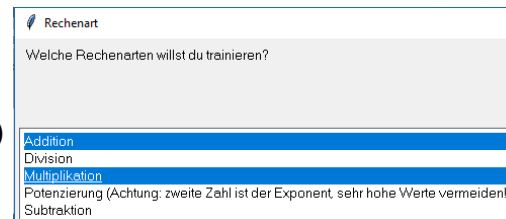
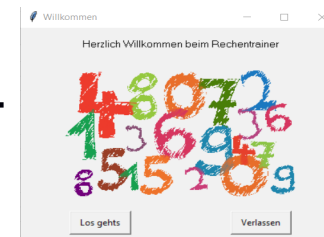
---

Featureliste, Ablaufbeispiele mit Belegungstabellen, Quellcode, Struktogramme



# Features

- Startseite mit Titelbild, Start- & Verlassenschaltfläche
- Automatisches Prüfen jeder Eingabe auf ungültige Eingaben (bspw. Text statt Zahl)
- 5 Rechenarten in beliebiger Kombination wählbar
  - (Additieren, Subtrahieren, Multiplizieren, Dividieren, Potenzieren)
  - Warnung vor zu großen Zahlen beim Potenzieren
- Freie Wahl beider Komponenten der Rechnungen (natürliche Zahlen)
- Wahl der Anzahl der zu rechnenden Aufgaben (ausreichend hohes Limit)
- Angabe aller Ergebnisse auf 2 Nachkommastellen
- 2 Versuche pro Aufgabe
  - Ein Punkt im ersten Versuch
  - Ein halber Punkt im zweiten Versuch
- Übersicht aller Aufgaben mit Nutzerergebnis am Ende
- Speichern der Übersicht möglich
- Direkte Rückkehr zur Startseite von überall mittels Cancel-Schaltfläche
- Verlassen ausschließlich durch Verlassen-Schaltfläche auf der Startseite möglich

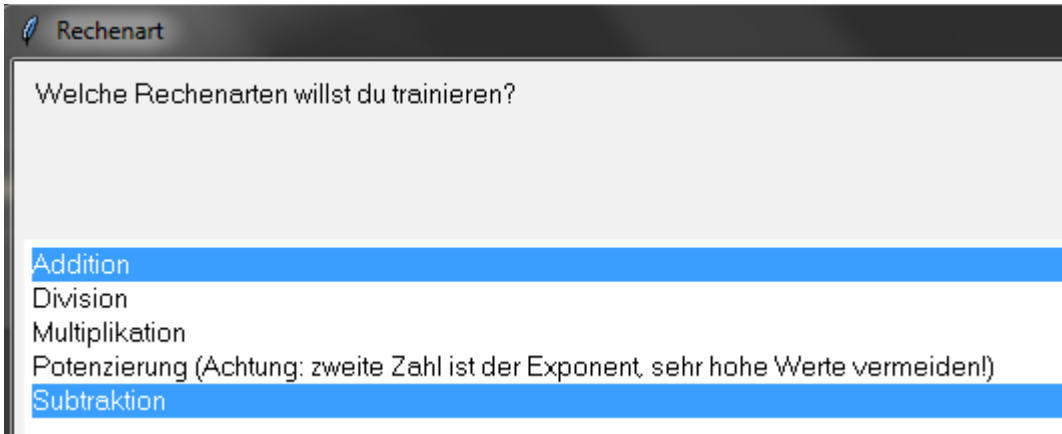


# Anwendungsbeispiele 1:

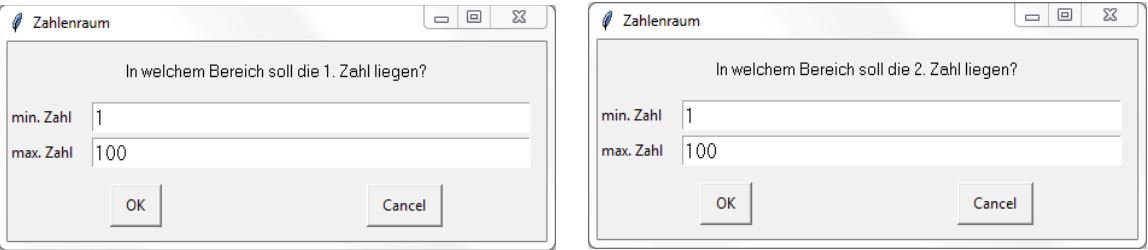
1. Johann öffnet den Rechentrainer und startet eine neue Runde



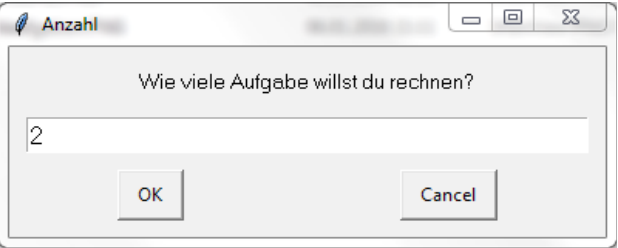
2. Da in seiner nächsten Arbeit Addition & Subtraktion bis 100 der Schwerpunkt sein werden wählt er diese beiden Rechenarten aus.



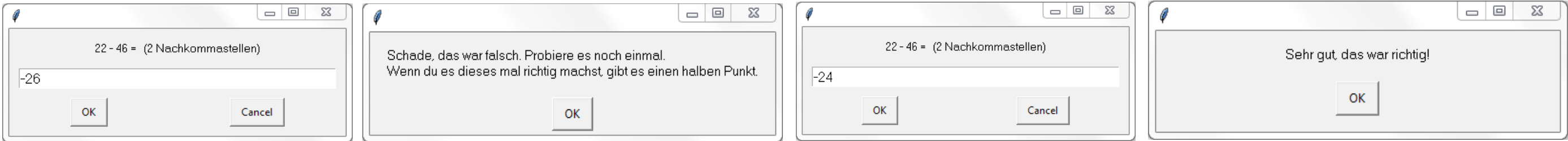
3. Danach legt er die Bereiche für beide Zahlen auf 1-100 fest.



4. Er hat nicht viel Zeit, daher möchte er nur 2 Aufgaben rechnen.



5. Seine erste Aufgabe (22-46) löst er zunächst falsch, im zweiten Versuch gibt er das richtige Ergebnis an. Er erhält einen halben Punkt.



Um zu prüfen, wie viele Versuche Johann noch hat, wird eine Schleife mit Eingangsbedingung gewählt, deren Zählervariable i kleiner als 2 sein muss um ausgeführt zu werden.

Ist die eingegebene Lösung falsch, wird i um 1 erhöht.

- 1. Bei zwei falschen Versuchen ist i also 2 und damit nicht kleiner als 2.
- 2. Bei einem richtigen Versuch wird i um 2 erhöht und ist somit direkt nicht mehr kleiner als 2.
- 3. Bei einem falschen und einem richtigen Versuch ist i 3 und somit auch nicht kleiner als 2.

In dieser Schleife befindet sich das Herzstück des Programms, die Anzeige der Aufgabe und die Eingabe des Ergebnisses.

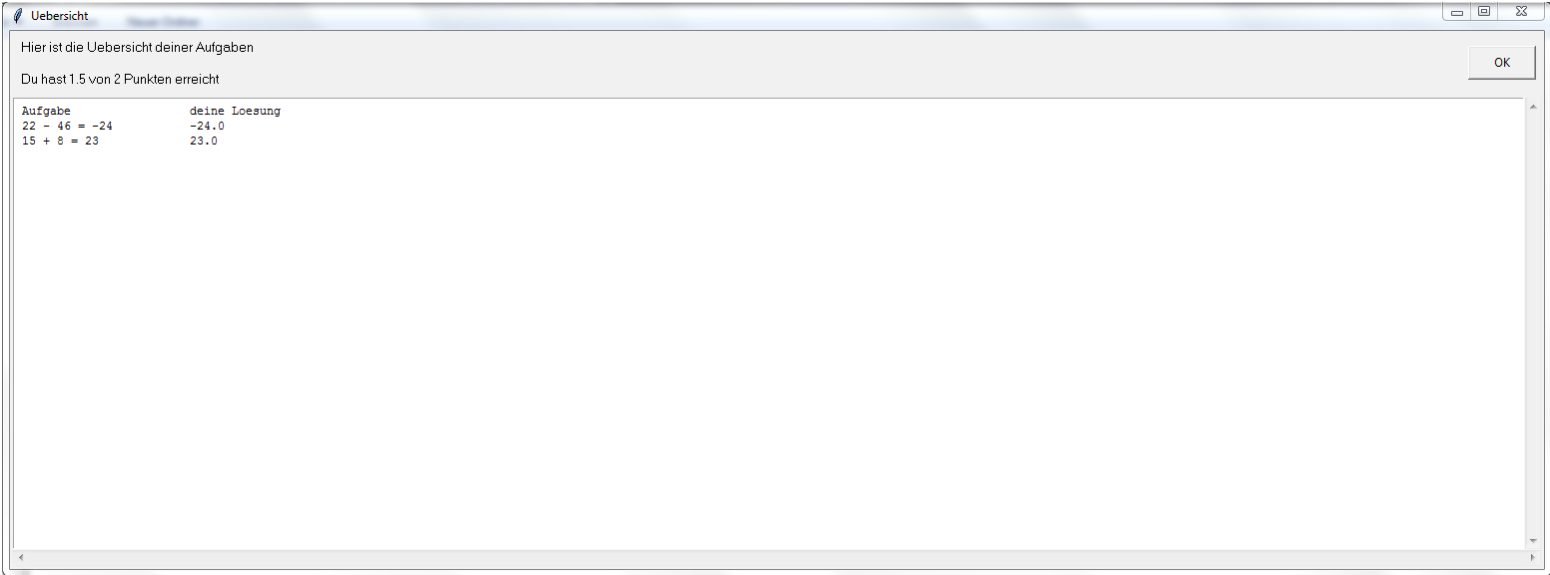
Zu Beginn wird die Punktzahl (richtige\_loesungen) immer um 1 erhöht. Bei jeder falschen Eingabe wird ein halber Punkt abgezogen, sodass bei zwei falschen Ergebnissen 0 Punkte zur Gesamtpunktzahl addiert werden. Die unten stehende Belegungstabelle stellt diese Sachverhalte für einen Durchlauf der Rechnung()-Funktion in Fall 3 dar.

Zeile	i	i < 2	richtige_loesungen	Ausgabe	Eingabe
206			1		
215	0				
217		WAHR			
224				22 - 46 = (2 Nachkommastellen)	
224					-26
250				Schade, das war falsch. [...]	
251			0,5		
252	1				
217		WAHR			
224				22 - 46 = (2 Nachkommastellen)	
224					-24
244				Sehr gut, das war richtig!	
245	3				
217		FALSCH			

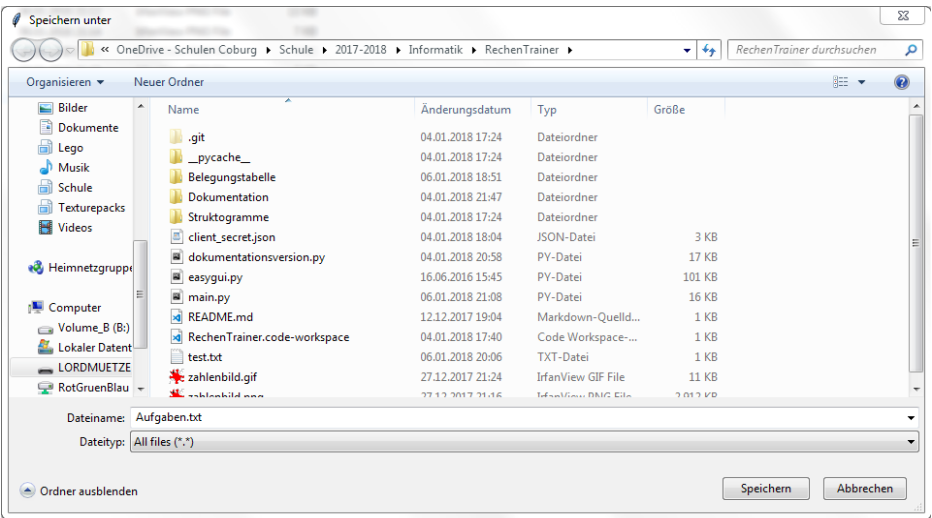
6. Die nächste Aufgabe (15+8) macht er auf Anhieb richtig und erhält einen Punkt.



7. Nun bekommt er eine Übersicht der soeben gerechneten Aufgaben inklusive seiner jeweils letzten Lösung angezeigt. Außerdem wird seine erreichte Punktzahl (1,5 von 2) ausgegeben.



8. Danach speichert er diese Übersicht als Aufgaben.txt ab und kehrt zur Startseite zurück.



9. Als letztes klickt er die Verlassen-Schaltfläche an und beendet so das Programm.

Während des gesamten Programmablaufs wird kontrolliert, ob Johann irgendwann eine Cancellschaltfläche anklickt. Sollte dies der Fall sein, ist der Funktionswert der jeweiligen Funktion ein `NoneType`, besitzt also keinen Wert. Beim Gesamtablauf, wird also bei jeder aufgerufenen Funktion mittels einer Variable namens `Funktionbezeichner_testen` geprüft, ob diese einen `NoneType` ausgibt. Sollte dies der Fall sein, so wird die Aufgabenschleife abgebrochen und zur Startseite zurück gekehrt. In diesem Beispiel klickt Johann keine Cancellschaltfläche an und `Funktionbezeichner_testen == None` ist somit immer `FALSCH`.

[illegible]