Lernatelier: Projektdokumentation

Benjamin Burgener

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 24.08.21 | 0.0.1 | Erste Version | Benjamin Burgener |
| 31.08.21 | 0.0.2 | Anpassung der Testfälle und  Realisieren bearbeiten | Benjamin Burgener |
| 07.09.21 | 0.0.3 | Allgemeine Anpassungen | Benjamin Burgener |
| 14.09.21 | 0.0.4 | Allgemeine Anpassungen | Benjamin Burgener |
| 21.09.21 | 1.0.0 | Finale Version | Benjamin Burgener |

# Informieren

## Ihr Projekt

Ich programmiere ein Programm, das automatisch eine Zahl zwischen 1 und 100 speichert.  
Der User kann dann eine Zahl raten.  
Nach jeder geratenen Zahl gibt der PC dem User einen Hinweis wie z.B das die Zahl höher oder tiefer ist und wenn es stimmt, dann kommt die Zahl wurde erraten  
Wenn die Zahl erraten wird dann werden die rateversuche aufgelistet.

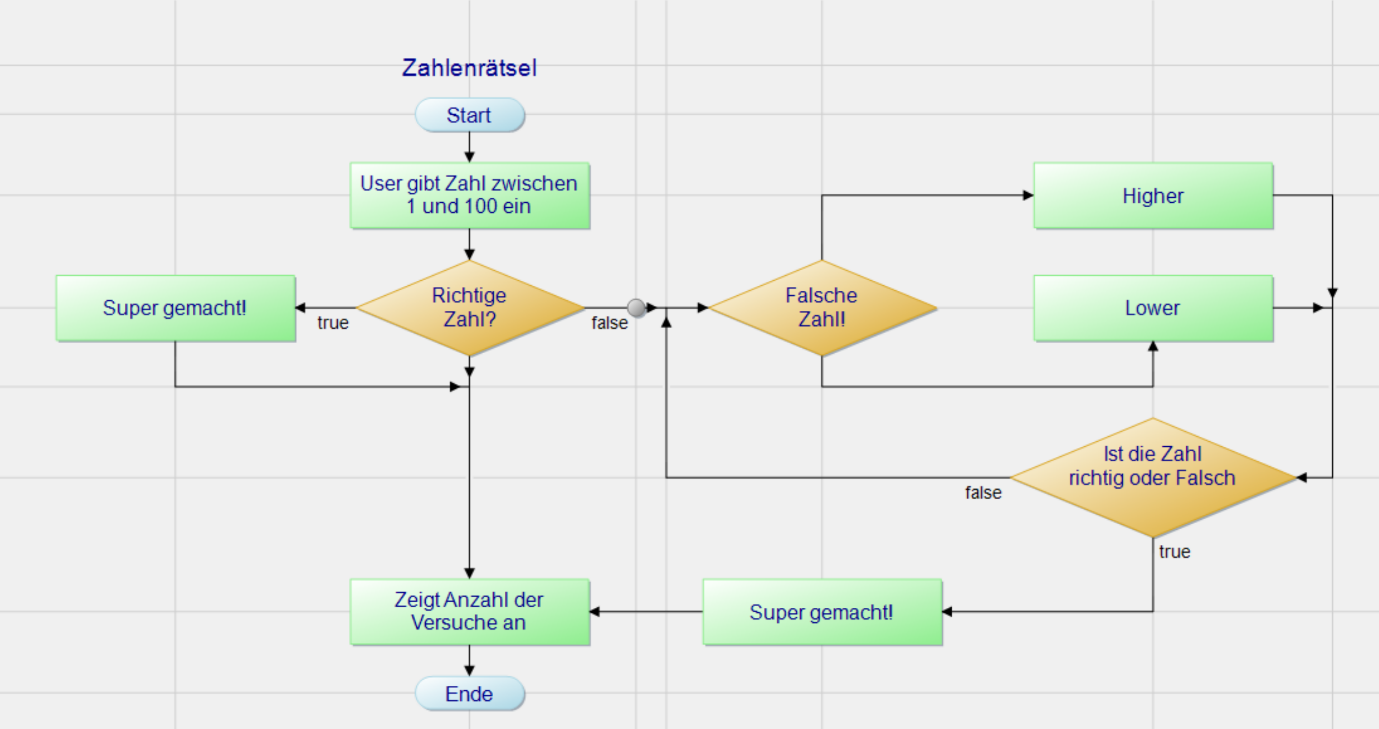
## Quellen

.net  
YouTube (Kanal: Programmieren Starten)

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1. | Muss | Funktional | Automatisch eine Zahl zwischen 1 und 100 auswählen können |
| 2. | Muss | Funktional | Der User kann eine Zahl raten |
| 3. | Muss | Funktional | Der Computer gibt Hinweise, ob die Zahl zu tief oder zu hoch war. |
| 4. | Muss | Qualität | Wenn die Zahl erraten ist, wird die Anzahl rate versuche angezeigt. |
| 5. | Kann | Qualität | Das Programm fragt nach, ob man nochmals spielen will. |
| 6. | Muss | Qualität | Das Programm kann fehleingaben umgehen oder vermeiden |
| 7. | Kann | Qualität | Eine Fehleingabe erscheint in roter Schrift |

## 1.4 Diagramme



1.5 Testfälle

[Erstellen Sie zu jeder Muss-Anforderung mindestens einen Testfall.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Vorbereitung** | **Eingabe** | **Erwartete Ausgabe** |
| 1. | IDE | Programm generiert eine Zahl zwischen 1-100 | Zufällige Zahl wird gespeichert |
| 2. | User gibt eine Zahl ein | Zahl zu klein oder zu gross | Antwort war richtig/falsch |
| 3. | Dem User wird gesagt, ob die Zahl zu gross oder zu klein war. | User gibt Zahl ein | Wenn die Zahl zu gross ist, soll der User eine kleinere eingeben und wenn die Zahl zu klein war soll er eine grössere eingeben. |
| 4 | Sobald der User die Zahl erraten hat, gibt das Programm an wie viele Versuche er gebraucht hat. |  | Erraten!!!  Sie haben die Zahl erraten  X versuche haben sie gebraucht |
| 5 | Das Programm soll sie Fehlereingabe umgehen oder vermeiden. | User gibt eine Fehleingabe ein (keine Zahl) | Error!  Geben sie bitte eine Zahl ein |

1. Planen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** |
| 1.1 | 31.8 | Ich lerne, wie man eine zufällige Zahl generiert. | 60 min |
| 1.2 | 31.8 | Der User kann eine Zahl eingeben. | 30 min |
| 1.3 | 7.9 | Der Computer kann Hinweise geben, ob die Zahl zu gross/klein war. | 60 min |
| 1.4 | 7.9 | Der Computer kann dem User aufzeigen, wie viel versuche er hatte. | 45 min |
| 1.5 | 14.9 | Der Computer kann die Fehlermeldung umgehen oder vermeiden. | 45 min |
| 1.6 | 14.9 | Das Programm fragt nach, ob man nochmals spielen will. | 20min |
| 1.7 | 14.9 | Das Programm gibt gewisse ausgaben rot an. | 20min |

1. Entscheiden

-

4. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| 1 | 31.8 | Ich lerne, wie man eine zufällige Zahl generiert. | 60min | 60min |
| 2 | 31.8 | Der User kann eine Zahl eingeben. | 30min | 15min |
| 3 | 7.9 | Der Computer kann Hinweise geben, ob die Zahl zu gross/klein war. | 60min | 45min |
| 4 | 7.9 | Der Computer kann dem User aufzeigen, wie viel versuche er hatte. | 45 min | 45min |
| 5 | 14.9 | Der Computer kann die Fehlermeldung umgehen oder vermeiden. | 45 min | 45min |
| 6 | 14.9 | Das Programm fragt nach, ob man nochmals spielen will. | 20min | 30min |
| 7 | 14.9 | Das Programm gibt gewisse ausgaben rot an. | 20min | 20min |

5. Kontrollieren

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 1 | 31.8 | **OK** | Burgener |
| 2 | 31.8 | **OK** | Burgener |
| 3 | 7.9 | **OK** | Burgener |
| 4 | 7.9 | **OK** | Burgener |
| 5 | 7.9 | **OK** | Burgener |
| 6 | 14.9 | **OK** | Burgener |
| 7 | 14.9 | **OK** | Burgener |

Die Tests haben weitgehend gut funktioniert und das Programm läuft einwandfrei.

6. Auswerten

Positiv:  
Es hat mir gefallen die bisher gelernten Sachen in diesem Auftrag zu nutzen und so etwas cooles zu Programmieren

Negativ:

Ich könnte bei den Tests mehr machen, aber ich denke das kommt mit der Zeit, das man immer mehr Testfälle schreiben muss, weil man sonst nicht mehr weiterkommt.