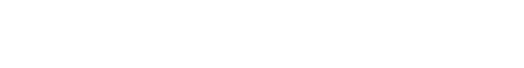
**Lernatelier: Projektdokumentation**



Nico Delvecchio

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
|  | 0.0.1 |  | Delvecchio |
|  |  | … |  |
| 2.9.2022 | 1.0.0 | Finale Version | Delvecchio |

# Informieren

1.1 Ihr Projekt

Ein Programm erstellen, das eine zufällige Zahlen generiert und der User diese erraten soll.

1.2 Quellen

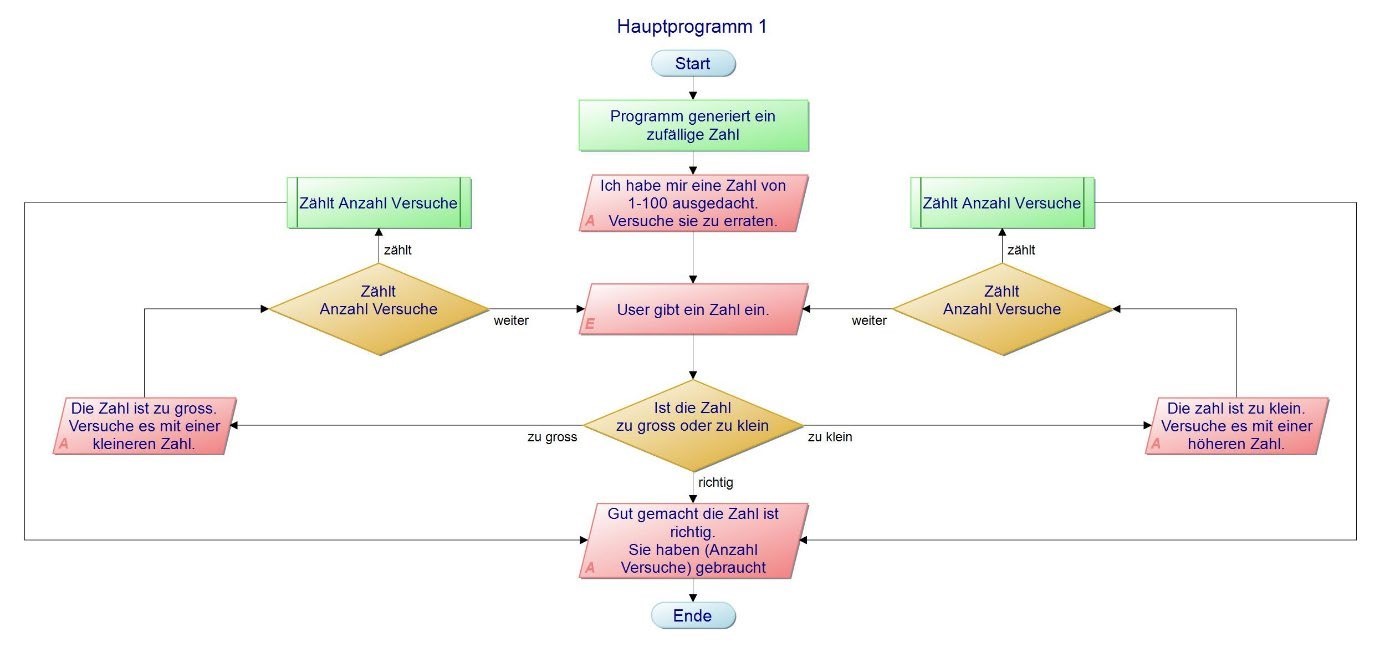
GitHub, YouTube, c-sharpcorner.com, stack-overflow und docs.microsoft.com

1.3 Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | muss | Funktional | Der Computer speichert eine Zufallszahl zwischen 1 bis 100 als Geheimzahl. |
| 2 | muss | Funktional | Benutzer soll die Zahl raten. |
| 3 | muss | Qualität | Für jede geratene Zahl etwas ausgeben   1. Die geratene Zahl ist niedriger als die Geheimzahl. 2. Die geratene Zahl ist grösser als die Geheimzahl. 3. Die Geheimzahl wurde erraten. |
| 4 | muss | Qualität | Wenn die Geheimzahl erraten wurde, soll die Anzahl der Rateversuche ausgegeben werden. |
| 5 | muss | Funktional | Das Programm soll mit Fehleingaben umgehen oder sie vermeiden können. |
| 6 | kann | Qualität | Erweiterungen, die über diese Anforderungen hinausgehen, sind möglich und willkommen. |

1.4

Diagramme



## Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| Nummer | Vorbereitung | Eingabe | Erwartete Ausgabe |
| 1 | Visual Studio starten und Programm läuft. | Auf Play klicken . | Zahl von 1-100 generieren und nicht anzeigen. |
| 2 | Programm gestartet und Zahl ist generiert. | Eingabe des Users, wenn sie richtig ist. | Dem User sagen, ob die Zahl stimmt, |
| 3.1 | Programm läuft, Zahl generiert | Zahl des Users ist kleiner. | Die geratene Zahl ist niedriger als die Geheimzahl. |
| 3.2 | Programm läuft, Zahl generiert | Zahl des Users ist grösser. | Die geratene Zahl ist  grösser als die Geheimzahl. |
| 3.3 | Programm läuft, Zahl generiert | Zahl des Users . | Die Geheimzahl wurde erraten. |
| 4 | Der User gibt seine Guess ein | Versuche des Users zählen. | Anzahl Rate versuche dem User anzeigen |
| 5 | Programm läuft, Alles funktioniert so weit | Eingabe des Users in Worten . | Worte nicht schreiben lassen und "Bitte nur Zahlen eingeben" ausgeben. |
| 6.1 | Visual Studio gestartet  Programm läuft und  Text ist vorhanden | Eingabe vom User. | Unter jedem Guess eine Linie zum Abgrenzen setzen. |
| 6.2 | Visual Studio gestartet  Programm läuft und  Text ist vorhanden | Eingabe des User | Text in verschiedenen Farben ausgeben |
| 6.3 | Visual Studio gestartet  Programm läuft und  Spiel ist beendet | Nachfrage an User, ob er weiterspielen will. Antwort des Users | Mit goto einen Code erstellt, damit der User weiterspielen kann oder das Programm endet. |
| 6.4 | Visual Studio gestartet  Programm läuft und  Fehleingabe des Users | Fehleingabe des Users | Mit Console.Beep(); einen Sound ausgeben, sobald eine Fehleingabe kommt, |

# Planen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** |
| 1 | - | Lernen wie ich eine Zufallszahle generiere und diese geheim behalte (nicht ausgeben) | 15-25 min |
| 2 | - | User gibt Zahl ein (Eingabe Code) | 10min |
| 3 | - | If/else code lernen und ausgeben können | 30 min |
| 4 | - | Count code lernen und am Ende ausgeben | 20 min |
| 5 | - | Dem User verweigern Worte schreiben zu lassen (Nur Zahlen). Den Code dazu lernen | 15 min |
| 6.1 | - | Änderungen am Code. Verschönerungen. | 3x 45 min |
| 6.2 | - | Ebenfalls Änderungen am Code vornehmen | 3 x 45 min |

# Entscheiden

Wie gehe ich vor?

Welche Quellen benutze ich?

Wie löse ich aufkommende Probleme?

Welche Textfarben benutze ich?

# Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Datum** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| 1.1 | 2.9.2022 | Random Number code geschrieben (ohne Ausgabe) | 5 | 3 |
| 2.1 | 2.9.2022 | Nach Eingabe gefragt und gespeichert | 5 | 5 |
| 3.1 | 2.9.2022 | Einen Boolean correct erstellt | 5 | 4 |
| 3.2 | 2.9.2022 | while schleife eingebaut | 10 | 8 |
| 3.2 | 2.9.2022 | If code für eine Zahl die zu hoch ist | 5 | 7 |
| 3.3 | 2.9.2022 | Else if für eine Zahl die zu niedrig ist | 5 | 3 |
| 3.4 | 2.9.2022 | Und einen Else code mit dem Boolean, falls die Zahl erraten wurde. | 10 | 13 |
| 4.1 | 2.9.2022 | Int tries erstellen und bei jedem if code tries++ reinschreiben | 5 | 2 |
| 4.2 | 2.9.2022 | Am Ende, wenn die Zahl erraten wurde, Anzahl versuch ausgeben. | 1 | 1 |
| 5.1 | 2.9.2022 | Mit Try.Parse konvertiert und einen code erstellt, damit wenn keine Zahl eingegeben wird, das Programm sagt, dass man eine Zahl eingeben soll. | 10 | 15 |
| 6.1 | 2.9.2022 | Unter jedem Versuch eine Linie ziehen, damit es übersichtlicher ist. | 2 | 1 |
| 6.2 | 2.9.2022 | Mit ConsoleColor die Farbe der Konsole geändert | 5 | 2 |
| 6.3 | 2.9.2022 | Mit goto einen Code geschrieben, der nach dem Spiel fragt, ob man weiterspielen will. | 10 | 15 |

# Kontrollieren

5.1 Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| **1** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **2** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **3** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **4** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **5** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **6** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **7** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **8** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **9** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **10** | 2.9.2022 | ✓ | **Nico Delvecchio** |
| **11** | 2.9.2022 | ✓ | **Nico Delvecchio** |

# Auswerten

# Positiv 🡪 Mein Arbeitsklima war gut.

# Ich hatte keine Probleme mit der Programmierung des Codes. Die Schleifen waren sehr schnell und gut geschrieben.

# Negativ 🡪 Ich muss mich mehr auf die Arbeit fokussieren, da ich schnell Ablenkbar bin.

# Beim Nachfragen, ob man weiterspielen will, habe ich eine go to Befehl verwendet, welcher sehr Fehleranfällig ist.