

Lernatelier: Projektdokumentation

Eslem Akdemir, Elisa Petit-Sirigu

Datum	Version	Änderung	Autor
11.01.2024	0.0.1	Informieren, Planen, Entscheiden	Eslem Akdemir, Elisa Petit-Sirigu
18.01.2024	0.0.2	Quellen, Planen	Elisa Petit-Sirigu
25.01.2024	0.0.3	Informieren	Eslem Akdemir, Elisa Petit-Sirigu
01.02.2024 08.02.2024	FERIEN		
29.02.2024	0.0.4	Realisieren, Kontrollieren	Eslem Akdemir, Elisa Petit-Sirigu
07.03.2024	1.0.0	Auswerten	Eslem Akdemir, Elisa Petit-Sirigu

Informieren

Ihr Projekt

Das Projekt ist eine individuelle Arbeit. Wir werden beide unsere OOP-Kenntnisse verbessern und üben objektorientiert zu programmieren. Die Zusammenarbeit besteht in der Unterstützung. Wir werden uns gegenseitig helfen und unterstützen. Allerdings werden wir an unterschiedlichem Code arbeiten und das auch beides auf GitHub veröffentlichen. Somit kann man unsere Entwicklung verfolgen.

Elisa Petit-Sirigu

Es ist ein Spiel mit 3 Würfeln. Sie können es mit mehreren Personen spielen. Am Anfang hat jeder Spieler 3 Chips und kann 3 Würfel werfen. Ein Spieler kann nur alle 3 Würfel werfen, wenn er/sie mindestens 3 Chips hat. Hat der Spieler einen oder zwei Chips, muss er/sie zwei Würfel werfen. Jeder Würfel bestimmt, was mit einem Chip passiert. Wenn du eine Eins würfelst, passiert nichts mit dem Chip. Bei einer Zwei passiert auch nichts mit dem Chip. Bei einer Drei passiert ebenfalls nichts mit dem Chip. Bei einer Vier musst du einen Chip nach links geben. Bei einer Fünf musst du einen Chip in die Mitte werfen. Bei einer Sechs musst du einen Chip nach rechts geben.

Nachdem du die Würfel geworfen hast, legst du deine Chips ab und gibst die Würfel nach rechts weiter.

[Repository](#)

Quellen

- <https://programmieren-starten.de/blog/objektorientierte-programmierung-in-csharp/>
-

Anforderungen Elisa Petit-Sirigu

Nummer	Muss / Kann?	Funktional? Qualität? Rand?	Beschreibung
1	Muss	Funktional	Das Programm muss laufen.
2	Muss	Funktional	Im Programm soll die Spiellogik implementiert werden. (Erklärung steht unter «Ihr Projekt bei Elisa Petit-Sirigu»)
3	Muss	Funktional	Das Programm soll den Benutzer fragen, wie der erste Spieler heisst.
4	Muss	Funktional	Der Benutzer soll Eingaben tätigen können.
5	Muss	Funktional	Nach der Eingabe des Benutzers, soll das Programm fragen, ob der Benutzer erneut einen Spieler hinzufügen möchte.
6	Muss	Funktional	Der Benutzer soll mit 1 oder 2 als ja und nein auf die Fragen antworten können.
7	Muss	Funktional	Das Programm soll den Benutzer so lange Fragen, ob er einen Spieler hinzufügen will, bis er nein klickt.
8	Muss	Funktional	Nach der Eingabe der Spieler soll das Spiel starten.
9	Muss	Funktional	Der Spielverlauf soll automatisch abgespielt werden.
10	Muss	Funktional	Am Ende muss der Gewinner angezeigt werden.
11	Muss	Funktional	Nach dem Spiel soll gefragt werden, ob der Benutzer erneut spielen möchte.
12	Muss	Funktional	Der Benutzer soll mit yes oder no das Spiel neustarten oder beenden können.
13	Muss	Funktional	Das Programm muss korrekt objektorientiert programmiert werde.

Anforderungen Eslem Akdemir

Nummer	Muss / Kann?	Funktional? Qualität? Rand?	Beschreibung
1	Muss	Funktional	Zusammenfassung über das Modul 320 schreiben
2	Muss	Funktional	In dieser Zusammenfassung werden Objekte beschrieben
3	Muss	Funktional	In dieser Zusammenfassung werden Klassen beschrieben
4	Muss	Funktional	In dieser Zusammenfassung werden Methoden beschrieben
5	Muss	Funktional	In dieser Zusammenfassung werden Attribute beschrieben
6	Muss	Funktional	In dieser Zusammenfassung wird UML beschrieben
7	Muss	Funktional	In dieser Zusammenfassung werden Instanzvariablen beschrieben
8	Muss	Funktional	In dieser Zusammenfassung wird die Kapselung beschrieben
9	Muss	Funktional	In dieser Zusammenfassung werden Konstruktoren beschrieben
10	Muss	Funktional	In dieser Zusammenfassung wird die Vererbung beschrieben

11	Muss	Funktional	In dieser Zusammenfassung werden Collections beschrieben
12	Muss	Funktional	In dieser Zusammenfassung werden Assoziationen beschrieben
13	Muss	Funktional	In dieser Zusammenfassung wird der Polymorphus beschrieben

Testfälle Elisa Petit-Sirigu

Nummer	Voraussetzung	Eingabe	Erwartete Ausgabe
1.1	Programm offen	Programm starten	Programm läuft ohne Fehler.
1.2	Spielanleitung gelesen	Programm starten	Programm benutzt die Korrekte Spiellogik
1.3	Programm offen	Programm starten	Konsole fragt den Benutzer wie der erste Spieler heisst.
1.4	Erste Frage in der Konsole wird angezeigt	Der Name: Emilia <i>enter</i>	Eingabe erfolgreich, das Spiel fährt fort.
1.5	Einen Spielernamen hinzugefügt	<i>enter</i>	Konsole fragt den Benutzer, ob er noch einen Spieler hinzufügen will.
1.6	Programm ist gestartet und man hat bereits einen Spieler eingegeben. Die Konsole fragt, ob man noch einen Spieler hinzufügen möchte.	1	Konsole fragt, wie der Spieler heisst.
1.6.1	Programm ist gestartet und man hat bereits einen Spieler eingegeben. Die Konsole fragt, ob man noch einen Spieler hinzufügen möchte.	2	Das Spiel startet automatisch
1.7	Benutzer hat bei der Frage bei 1.6 1 geklickt und gibt den Namen ein.	Jonas	Die Konsole fragt, ob der Benutzer einen weiteren Spieler hinzufügen möchte.
1.8	Alle Spieler wurden angegeben	<i>enter</i>	Spiel startet automatisch.
1.9	Spiel hat automatisch gestartet	Keine Eingabe	Das Spiel läuft von allein und macht alle Spielzüge.
1.10	Spiel läuft automatisch	Keine Eingabe	Gewinner wird automatisch angezeigt.
1.11	Gewinner wird angezeigt	Keine Eingabe	Konsole fragt, ob man noch eine Runde spielen möchte.

1.12	Konsole fragt, ob man noch eine Runde spielen möchte.	yes	Spiel startet von neu
1.12.1	Konsole fragt, ob man noch eine Runde spielen möchte.	no	Programm schliesst sich

Testfälle Eslem Akdemir

Nummer	Voraussetzung	Eingabe	Erwartete Ausgabe
1.1	Zusammenfassung ist geöffnet	-	-
2.1	Zusammenfassung ist geöffnet	Kapitel «Objekte» öffnen	-
3.1	Zusammenfassung ist geöffnet	Kapitel «Klassen» öffnen	-
4.1	Zusammenfassung ist geöffnet	Kapitel «Methoden» öffnen	-
5.1	Zusammenfassung ist geöffnet	Kapitel «Attribute» öffnen	-
6.1	Zusammenfassung ist geöffnet	Kapitel «Klassen, Attribute, Methoden und Objekte im UML» und «Attribute UML» und «Konstruktoren UML» und «UML Sequenzdiagramme» öffnen	-
7.1	Zusammenfassung ist geöffnet	Kapitel «Instanzvariablen» öffnen	-
8.1	Zusammenfassung ist geöffnet	Kapitel «Kapselung» öffnen	-
9.1	Zusammenfassung ist geöffnet	Kapitel «Objekt Instanziierung und Konstruktoren» öffnen	-
10.1	Zusammenfassung ist geöffnet	Kapitel «Vererbung» öffnen	-
11.1	Zusammenfassung ist geöffnet	Kapitel «Collections» öffnen	-
12.1	Zusammenfassung ist geöffnet	Kapitel «Assoziationen» öffnen	-
13.1	Zusammenfassung ist geöffnet	Kapitel «Polymorphus» öffnen	-

Planen

Nummer	Frist	Beschreibung	Zeit (geplant)
1.	11.01	Projektwahl treffen und Dokumentation bis und mit entscheiden fertigstellen.	2 * 5
2.	18.01	Geeigneten Kurs für das Lernen der objektorientierten Programmierung finden.	2 * 2
3.	24.01	Projektbericht schreiben.	-
4.	25.01	Sich mit den Grundlagen der objektorientierten Programmierung befassen.	2 * 5
5.	25.01	Projektbericht schreiben.	-
6.	01.02	Kleines Programm in C# und objektorientierten programmieren.	2 * 5
7.	01.02	Dokumentation Realisieren und Testprotokoll ausfüllen.	2 * 1
8.	21.02	Projektbericht schreiben. Frist 18.00	-
9.	22.02	Sich mit fortgeschritteneren Aufgaben beschäftigen.	2 * 6
10.	22.02	Projektbericht schreiben.	-
11.	29.02	Fortgeschrittenes Programm schreiben.	2 * 5
12.	29.02	Dokumentation Realisieren und Testprotokoll ausfüllen.	2 * 1
13.	06.03	Projektbericht schreiben. Frist 18.00	-
14.	07.03	Dokumentation Kontrollieren und Auswerten ausfüllen.	2 * 1
15.	07.03	Portfolio schreiben.	2 * 4

35 Lektionen pro Person.

Der Projektantrag muss nicht in den Lernatelierlektionen geschrieben werden, weshalb wir ihn nicht in die Planung mit einbezogen haben.

Entscheiden

Elisa Petit-Sirigu

- Während dem Ablauf des Projekts, habe ich mich dazu entschieden nicht zwei, sondern nur ein OOP-Programm zu programmieren, weil ich keine Ideen für ein kleines Projekt hatte. Deshalb habe ich, als mein «kleines Projekt» den Grundaufbau gemacht und als grosses Projekt mein Gerüst ergänzt.

Realisieren

Nr.	Frist	Beschreibung	Zeit (geplant)	Zeit (effektiv)
1.	11.01	Projektwahl treffen und Dokumentation bis und mit entscheiden fertigstellen.	2 * 5	2 * 5
2.	18.01	Geeigneten Kurs für das Lernen der objektorientierten Programmierung finden.	2 * 2	2 * 2
3.	24.01	Projektbericht schreiben.	-	-
4.	25.01	Sich mit den Grundlagen der objektorientierten Programmierung befassen.	2 * 5	2 * 5
5.	25.01	Projektbericht schreiben.	-	-
6.	01.02	Kleines Programm in C# und objektorientierten programmieren.	2 * 5	2 * 5
7.	01.02	Dokumentation Realisieren und Testprotokoll ausfüllen.	2 * 1	2 * 1
8.	21.02	Projektbericht schreiben. Frist 18.00	-	-
9.	22.02	Sich mit fortgeschritteneren Aufgaben beschäftigen.	2 * 6	2 * 6
10.	22.02	Projektbericht schreiben.	-	-
11.	29.02	Fortgeschrittenes Programm schreiben.	2 * 5	2 * 5
12.	29.02	Dokumentation Realisieren und Testprotokoll ausfüllen.	2 * 1	2 * 1
13.	06.03	Projektbericht schreiben. Frist 18.00	-	-
14.	07.03	Dokumentation Kontrollieren und Auswerten ausfüllen.	2 * 1	2 * 1
15.	07.03	Portfolio schreiben.	2 * 4	2 * 4

Kontrollieren

Testprotokoll Elisa Petit-Sirigu

Testumgebung: Visual Studio Code

Computer: Windows 11

Nummer	Test Nr.	Datum	Resultat	Durchgeführt
1.	1.1	07.03.2024	OK	Elisa Petit Sirigu
2.	1.2	07.03.2024	OK	Elisa Petit Sirigu
3.	1.3	07.03.2024	OK	Elisa Petit Sirigu
4.	1.4	07.03.2024	OK	Elisa Petit Sirigu
5.	1.5	07.03.2024	OK	Elisa Petit Sirigu
6.	1.6	07.03.2024	OK	Elisa Petit Sirigu
7.	1.6.1	07.03.2024	OK	Elisa Petit Sirigu
8.	1.7	07.03.2024	OK	Elisa Petit Sirigu
9.	1.8	07.03.2024	OK	Elisa Petit Sirigu
10.	1.9	07.03.2024	OK	Elisa Petit Sirigu
11.	1.10	07.03.2024	OK	Elisa Petit Sirigu
12.	1.11	07.03.2024	OK	Elisa Petit Sirigu
13.	1.12	07.03.2024	OK	Elisa Petit Sirigu
14.	1.12.1	07.03.2024	OK	Elisa Petit Sirigu

Testprotokoll Eslem Akdemir

Testumgebung: Visual Studio Code

Computer: Windows 10

Nummer	Test Nr.	Datum	Resultat	Durchgeführt
1.1.1	1.1	07.03.2024	OK	Eslem Akdemir
2.1.1	2.1	07.03.2024	OK	Eslem Akdemir
3.1.1	3.1	07.03.2024	OK	Eslem Akdemir
4.1.1	4.1	07.03.2024	OK	Eslem Akdemir
5.1.1	5.1	07.03.2024	OK	Eslem Akdemir
6.1.1	6.1	07.03.2024	OK	Eslem Akdemir
7.1.1	7.1	07.03.2024	OK	Eslem Akdemir
8.1.1	8.1	07.03.2024	OK	Eslem Akdemir
9.1.1	9.1	07.03.2024	OK	Eslem Akdemir
10.1.1	10.1	07.03.2024	OK	Eslem Akdemir
11.1.1	11.1	07.03.2024	OK	Eslem Akdemir
12.1.1	12.1	07.03.2024	OK	Eslem Akdemir
13.1.1	13.1	07.03.2024	OK	Eslem Akdemir

Auswerten

Elisa Petit-Sirigu:

Wie ist das Projekt gelaufen?

Dieses Projekt war sehr speziell aufgebaut. Weil ich mit meinem Teammitglied nur sehr wenig Kontakt während dem Projekt hatte, fühlte es sich meisten wie ein Einzelprojekt an.

Ich kam aber gut voran. Der Plan war, ein Kleineres Programm zu machen, bei dem man die Basics wieder repetieren kann und danach ein grösseres, um sich in das OOP zu vertiefen. Allerdings ist mir nichts Gutes für ein kleines Projekt eingefallen. Deshalb habe ich, als mein «kleines Projekt» den Grundaufbau gemacht und als grosses Projekt mein Gerüst ergänzt.

Diese Aufteilung hat für mich gut funktioniert. In diesem Projekt ist mir die Planung sehr gut gelungen. Ich musste nichts verschieben, war immer im Zeitplan und hatte am Ende sogar noch Zeit übrig. Das ist das erste Projekt, bei dem alles so gelaufen ist, wie ich es geplant habe.

Eslem Akdemir:

Ich konnte alle Anforderungen erreichen und somit ist das Projekt vollständig.