

---

# OBJEKTORIENTIERT PROGRAMMIEREN

---

Modul 320



Figure 1



6. JULI 2023  
MIRALEM BANOVI  
Version 1

# Inhalt

<b>Informieren</b> .....	3
Auftrag.....	3
<i>Problemstellung:</i> .....	3
Anforderungen: .....	3
<i>Programm</i> .....	3
<i>Dokumentation</i> .....	4
<b>Planen</b> .....	5
Analyse .....	5
Ausgangslage (Ist- Situation) .....	5
Projektziel (Soll-Situation) .....	5
Wieso macht man Karteikarten? .....	5
Scoring-System .....	6
Anforderung .....	6
Funktional.....	6
Nichtfunktional .....	6
Funktion des Projekts .....	7
<b>Realisieren</b> .....	8
User Stories .....	8
Rollen.....	8
Stories.....	8
<b>Entscheiden</b> .....	9
Programmbeschreibung .....	9
Hauptmenü.....	9
Startbildschirm .....	9
Ordnerauswahl .....	10
Lernprozess.....	10
Score .....	11
Ende.....	11
Klassendiagramm .....	12
MainWindow:.....	12
StartExit: .....	12
OrdnerAuswahl:.....	12
QuestionWindow:.....	13
ScoreExit: .....	13
<b>Kontrollieren</b> .....	19

Testing.....	19
Teststrategie .....	19
Testumgebung .....	19
Unit-Testing.....	22
Arbeitsjournal.....	23
<b>Auswerten</b> .....	26
Fazit .....	26
Persönliches Fazit: .....	26
<b>Quellen</b> .....	27
<b>Abbildungsverzeichniss</b> .....	29
<b>TabellenVerzeichniss</b> .....	29

## Informieren

### Auftrag

Wir wurden beauftragt, eine Fallstudie zu einer Problemstellung durchzuführen und eine objektbasierte Applikation basierend auf den Ergebnissen zu entwickeln. In der Fallstudie müssen Sie das vorliegende Problem analysieren und eine passende Lösung entwerfen und modellieren. Dabei sollten Sie Überlegungen zur Implementierung und Gestaltung der Benutzerschnittstelle anstellen. Die genaue Funktionsweise der Lösung muss in der Analyse eigenständig erarbeitet werden. Hier sind die grundlegenden Anforderungen:

### *Problemstellung:*

Die meisten kostenlosen Karteikarten-Programme bieten nur begrenzte Funktionen zum Erstellen von Karteikarten mit verschiedenen Medien (Bild, Text und Ton). Das effektive Üben von Vokabeln oder Fachbegriffen mit unterschiedlichen Medien ist daher eingeschränkt.

In dieser Fallstudie soll eine neue Lösung entwickelt werden, bei der der Benutzer Bilder, Texte und Audiodateien wild durcheinander als Karteikarten verwenden kann.

Beispiel: Angenommen, der Benutzer möchte englisches Vokabular lernen. Das Programm sollte dem Benutzer Karteikarten anzeigen können, die entweder ein Bild, ein deutsches Wort oder eine Audiodatei enthalten. Der Benutzer muss dann das Lösungswort erkennen und beispielsweise in einem Textfeld zur Überprüfung eingeben. Die Art der Überprüfung kann auch auf andere Weise erfolgen. Solche Details müssen in der Fallstudie erarbeitet werden, um ein Programm zu entwickeln, das den Benutzer aktiv beim Lernen unterstützt. Möglicherweise könnte ein Scoring-System implementiert werden.

Der Benutzer sollte in der Lage sein, "Hinweisdateien" (also Bilder, Texte oder Audiodateien) selbst zum Programm hinzuzufügen. Dadurch kann der Benutzer seine eigene Sammlung von Karteikarten erstellen, aus der das Programm dem Benutzer dann Karteikarten in zufälliger Reihenfolge anzeigen kann.

### Anforderungen:

#### *Programm*

Der Benutzer kann Karteikarten mit Bildern, Texten oder Audiodateien erstellen.

Das Programm unterstützt den Benutzer aktiv beim Lernen.

Der Benutzer kann die Hinweisdateien und Lösungswörter selbst zum Programm hinzufügen.

Das Programm prüft die Lösung des Benutzers und gibt Feedback zur Korrektheit der Eingabe.

## *Dokumentation*

Es muss eine umfassende Dokumentation zum gesamten Projekt (einschliesslich der Fallstudie) geführt werden.

Das Problem muss vor der Umsetzung in der Dokumentation analysiert werden.

Basierend auf der Analyse wird ein Pflichtenheft mit allen Anforderungen an das Endprodukt erstellt.

Die Dokumentation muss den gewählten Lösungsansatz in allen Bereichen vollständig beschreiben und begründen.

Alle Aspekte des Programms müssen detailliert in der Dokumentation beschrieben werden. Was nicht in der Dokumentation steht, existiert nicht und wird nicht bewertet.

Die Klassenstruktur muss im Voraus anhand eines Klassendiagramms geplant werden.

Die Dokumentation muss alle relevanten User Stories für das Programm enthalten.

Vor der Umsetzung müssen Mockups für die Benutzerschnittstelle erstellt werden, in denen alle Ansichten und Aktionen eindeutig beschrieben sind.

Es ist erforderlich, ein ausführliches Arbeitsjournal zu führen.

9.1. Im Arbeitsjournal werden alle Tätigkeiten am Projekt und die benötigte Arbeitszeit für jeden Arbeitstag aufgelistet.

9.2. Am Ende jedes Arbeitstages wird eine Reflexion geschrieben, die die Ereignisse des Tages, Probleme und Lernerfahrungen in ein paar Sätzen zusammenfasst.

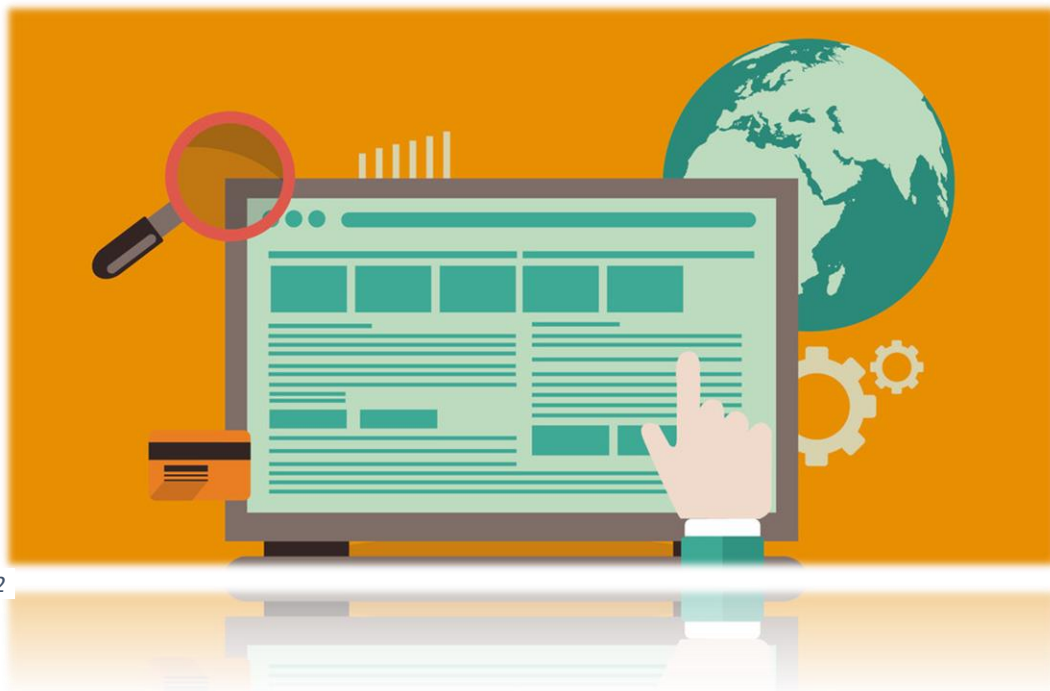


Figure 2

## Planen

### Analyse

#### Ausgangslage (Ist- Situation)

Wenn das Projekt fertig ist, wird eine umfassende Lösung zur Verfügung stehen. Durch die Dokumentation werden alle Schritte und Abläufe des Projekts transparent. Die Bedienung des Programms wird für jeden Benutzer mühelos sein, sodass ohne grossen Aufwand Karteikarten erstellt werden können, unabhängig davon, ob es sich um Text oder Bilder handelt. Mit Hilfe des Projekts wird es wesentlich einfacher sein, Wörter, Sprachen, Hauptstädte und andere Dinge zu lernen. Die Anwendung wird eine wertvolle Unterstützung bieten, um das Lernen effektiver und effizienter zu gestalten, und gleichzeitig eine benutzerfreundliche Erfahrung für alle bieten.

Die Anwendung wird dir helfen, dein Wissen zu vertiefen und dich gleichzeitig motivieren, weiterzumachen.

Es ist ratsam, sich Bilder nicht auswendig zu merken, da dies zu Fehlern führen kann. Es ist besser, auf den Bildern Merkmale wie die Automarke zu erkennen, anstatt nur zu sagen, dass das Auto rund ist.

#### Projektziel (Soll-Situation)

Das Projektziel, also die Soll-Situation, meines Projekts ist es, eine benutzerfreundliche Anwendung zu entwickeln, die das Lernen mithilfe von interaktiven Lernkarten effektiver und unterhaltsamer gestaltet. Die Anwendung soll es den Benutzern ermöglichen, Karteikarten mit verschiedenen Inhalten wie Texten und Bildern zu erstellen und diese in einer interaktiven Umgebung zu verwenden. Die Hauptziele meines Projekts sind die einfache Bedienbarkeit der Anwendung, die Vielseitigkeit der Karteikarten, die Förderung des effektiven Lernens, die Motivation und der Spass beim Lernen, die Flexibilität des Lernens, die Verfolgung des Lernfortschritts und die Möglichkeit zur Kollaboration und zum Austausch von Lernmaterialien zwischen Benutzern. Letztendlich strebt das Projektziel danach, eine Lernplattform zu schaffen, die das Lernen erleichtert, die Motivation steigert und eine angenehme Lernerfahrung bietet, während sie gleichzeitig den Fortschritt der Benutzer verfolgt und unterstützt.

### Wieso macht man Karteikarten?

Karteikarten sind für mich eine effektive Methode, um Informationen zu lernen und mich an Wissen zu erinnern. Wenn ich regelmässig durch die Karten gehe, kann ich die Inhalte besser im Gedächtnis behalten. Die Karteikarten ermöglichen es mir, Informationen auf das Wichtigste zu reduzieren. Ich kann wichtige Fakten, Definitionen oder Zusammenhänge auf kleinem Raum notieren. Die Karteikarten sind portabel und flexibel. Ich kann sie überallhin mitnehmen und immer mal wieder kurz üben, sei es in der Bahn, in der Pause oder zu Hause. Mit den Karteikarten kann ich mich selbst abfragen. Dies fördert mein aktives Lernen und hilft mir, das Wissen besser zu verinnerlichen. Die Karteikarten sind hilfreich, um mich auf Prüfungen oder Tests vorzubereiten. Ich kann wichtige Informationen strukturiert und übersichtlich festhalten und so meinen Lernprozess unterstützen. Beim Erstellen der Karteikarten kann ich die Informationen in meinen eigenen Worten formulieren, was mein Verständnis fördert, und mein kreatives Denken anregt. Wenn ich die Karteikarten nach dem Prinzip des "aktiven Rückrufs" verwende, kann ich gezielt die Inhalte üben, bei denen ich Schwierigkeiten habe. Ich kann die Karteikarten thematisch oder nach Schwierigkeitsgrad sortieren, was die Organisation meines Lernstoffs erleichtert. Insgesamt sind Karteikarten für mich also eine vielseitige Lernmethode, die sich für verschiedene Situationen und meinen Lerntyp eignet. Sie unterstützen mein aktives Lernen, fördern meine Erinnerung und erleichtern das gezielte Üben von Inhalten.

## Scoring-System

Das Scoring-System in meinem Projekt bewertet meine Leistung, während ich die Anwendung nutze. Wenn ich Lernkarten betrachte und meine Antworten darauf gebe, sammle ich Punkte basierend auf meinen Antworten.

Wenn ich eine Lernkarte anschau und meine Antwort eingebe, kann ich auf den "Überprüfen"-Button klicken. Das Scoring-System vergleicht dann meine Antwort mit der richtigen Antwort. Wenn meine Antwort korrekt ist, erhalte ich Punkte, die zu meinem Gesamtscore hinzugefügt werden. Mein Score wird angezeigt und aktualisiert, während ich weitere Lernkarten bearbeite.

Mein Ziel ist es, meinen Score zu erhöhen, indem ich so viele richtige Antworten wie möglich gebe. Je mehr Lernkarten ich korrekt beantworte, desto höher wird mein Score. Das Scoring-System motiviert mich, weiterzumachen und mich beim Lernen zu verbessern.

Am Ende werden mir meine gesammelten Punkte angezeigt, nachdem ich alle verfügbaren Lernkarten durchgearbeitet habe. Das ermöglicht es mir, meine Leistung zu sehen und meinen Lernfortschritt zu verfolgen.

## Anforderung

### Funktional

	Anforderung
1	Karteikarten haben zufällige Reihenfolge
2	Als Medium kann Bild oder Text ausgewählt werden
3	Bei der Eingabe falsche Antwort kommt man nicht weiter
4	Die Anwendung ist einfach zu bedienen

*Tabelle 1*

*Ref.Sollberger Doku*

### Nichtfunktional

	Anforderung
1	Sollte dich beim Lernen unterstützen
2	Das Programm enthält einfache GUI
3	Responsivität und Plattformunabhängigkeit
4	Leistung und Reaktionszeit

*Tabelle 2*

*Ref.Sollberger Doku*

## Funktion des Projekts

Dieses Projekt soll ein interaktives Lernerlebnis bieten, bei dem Benutzer mehr über verschiedene Automarken erfahren können. Alle relevanten Inhalte wie Bilder, Texte und Audiodateien werden in einem bestimmten Ordner organisiert. Benutzer haben die Möglichkeit, diesen Ordner auszuwählen und das Lernspiel zu starten.

Das Hauptfenster, „MainWindow“ genannt, begrüßt den Benutzer freundlich und bietet einen Button namens „Start“, um das Spiel zu starten. Nachdem auf „Start“ geklickt wurde, öffnet sich ein weiteres Fenster namens „StartExit“. Hier wird eine kurze Erläuterung zur Bedeutung von Karteikarten in der digitalen Welt gegeben. Benutzer können dann auf „Weiter“ klicken, um den Ordner auszuwählen, in dem die Lernmaterialien gespeichert sind.

Die Klasse „FolderSelect“ ermöglicht es dem Benutzer, den spezifischen Ordner auszuwählen, in dem die Lernmaterialien gespeichert werden. Sobald der Benutzer den Ordner ausgewählt hat, schließt sich dieses Fenster automatisch und das eigentliche Spiel beginnt.

Im „QuestionWindow“ wird den Nutzern ein Bild, Text oder Audio präsentiert und sie werden gebeten, den Inhalt zu identifizieren. Benutzer haben die Möglichkeit, ihre Antwort in ein Textfeld einzugeben und dann auf „Bestätigen“ zu klicken. Bei richtiger Antwort werden neue Inhalte angezeigt. Wenn nicht, wird der aktuelle Bildschirm beibehalten, um einen erneuten Versuch zu ermöglichen.

Insgesamt bietet dieses Projekt eine interaktive Möglichkeit, Wissen über Automarken zu testen und zu erweitern. Die benutzerfreundliche Oberfläche fördert ein angenehmes Lernerlebnis.

Folgende Kriterien sind für die Struktur gültig, welche den Text gestaltet.

- Im Kartei-Ordner befindet sich mindestens ein Ordner
- Im Kartei-Ordner befindet sich keine Datei
- In jedem Karteikarten-Ordner befindet sich mindestens ein Bild oder Text
- Erlaubte Dateiformate für  
Bilder = JPG und PNG  
Texte = TXT

Sobald die Struktur korrekt ist, erfolgt das Laden der Karteikarten in eine Liste. Der Benutzer wird mit einem Startbildschirm begrüßt, auf dem die Anzahl der geladenen Karteikarten angezeigt wird. Von hier aus kann der Lernprozess gestartet werden. Ref.Sollberger

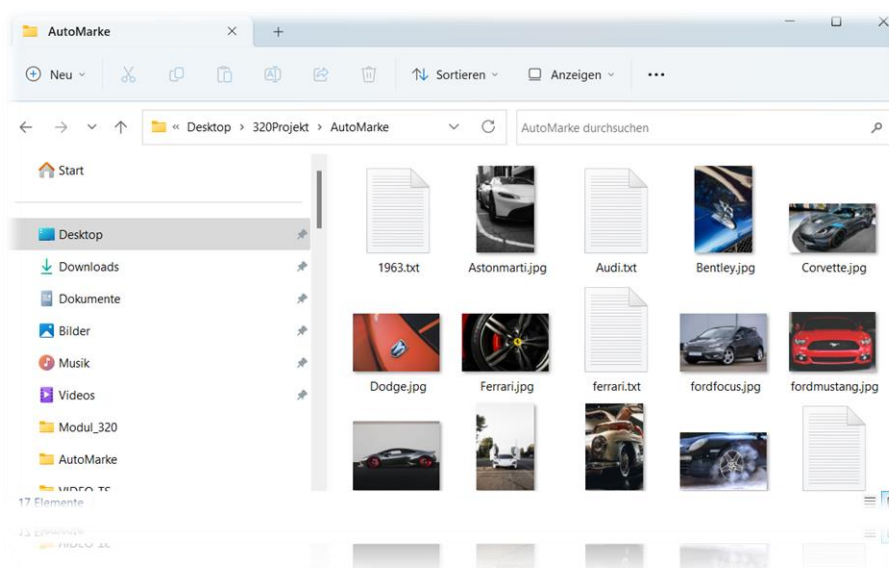


Figure 3



## Realisieren

### User Stories

Ich möchte die Anwendung starten können, um meine Lernkarten besser zu organisieren und zu nutzen. Ich will verschiedene Funktionen haben, um Lernkarten auszuwählen, meine Antworten zu überprüfen und meinen Score zu verfolgen. In der Anwendung sehe ich ein Hauptfenster mit einem Start-Button. Wenn ich auf den Start-Button klicke, erscheint das Hauptmenü mit einem "Weiter"-Button und einer Erklärung zu Lernkarten. Wenn ich auf "Weiter" klicke, kann ich einen Ordner mit meinen Lernkarten auswählen. Die Anwendung zeigt mir dann die Lernkarten in zufälliger Reihenfolge an.

Für jede Lernkarte gebe ich meine Antwort ein und klicke auf "Überprüfen", um zu sehen, ob meine Antwort richtig ist. Die Anwendung sagt mir, ob meine Antwort korrekt ist und erhöht meinen Score, wenn ich richtig antworte. Wenn verfügbar, wird die nächste Lernkarte angezeigt. Ich kann die Anwendung schliessen und später weitermachen, um meine Lernkarten zu nutzen. Die Anwendung ermöglicht es mir, meine Lernkarten einfach zu organisieren und zu nutzen, indem sie eine effiziente Möglichkeit bietet, meine Antworten zu überprüfen und meinen Fortschritt zu verfolgen.

### Rollen

In dieser Anwendung gibt es zwei Hauptrollen: der Lernende und der Anwender. Als Lernender verwende ich die Anwendung, um meine Lernkarten besser zu organisieren und meinen Fortschritt zu verfolgen. Ich wähle Lernkarten aus, beantworte Fragen und erhalte sofortiges Feedback zu meinen Antworten. Mein Ziel ist es, meinen Score zu verbessern, indem ich die richtigen Antworten finde. Der Anwender ist der Entwickler oder Administrator der Anwendung. Er erstellt die Benutzeroberfläche und implementiert die Funktionalitäten, die es mir als Lernendem ermöglichen, die Lernkarten effektiv zu nutzen. Der Anwender sorgt dafür, dass die Anwendung intuitiv zu bedienen ist und reibungslos funktioniert, um meine Lernerfahrung zu optimieren.

### Stories

	User Story
1	Als Benutzer kann ich die Karteikarten durchgehen
2	Als Benutzer kann ich meine Antwort vom Programm kontrollieren lassen, um zu sehen, ob ich den Begriff richtig gelernt habe.
3	Als Benutzer kann ich am Ende des Spiels das Spiel neustarten
4	Als Benutzer kann ich Karteikarten erstellen

Tabelle 3

Ref.Sollberger Doku

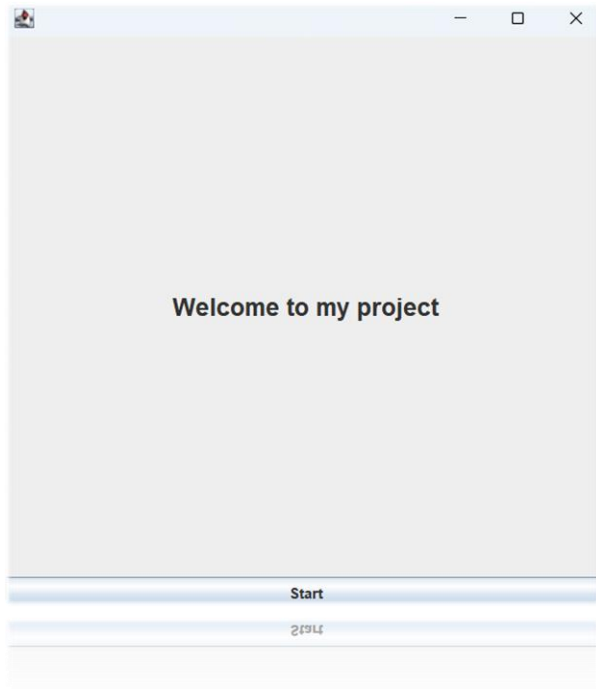
## Entscheiden

### Programmbeschreibung

#### Hauptmenü

Sobald ich mein Projekt starte, erscheint als erstes Fenster eine Begrüss und eine Nachricht mit dem Titel "Willkommen bei meinem Projekt". Wenn ich auf die Schaltfläche "Start" klicke, öffnet sich das darauffolgende Fenster. >

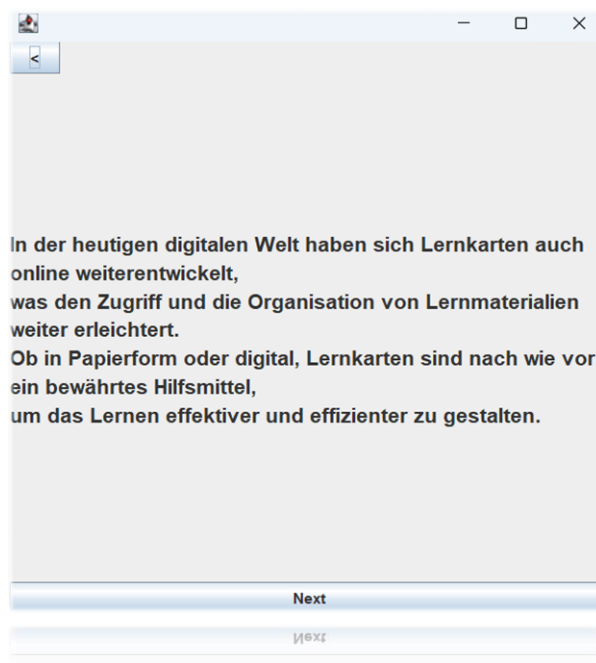
Figure 4



#### Startbildschirm

Anschliessend wird dieses Fenster angezeigt, begleitet von einem Willkommenstext. Bitte klicken Sie auf "Weiter".

Figure 5



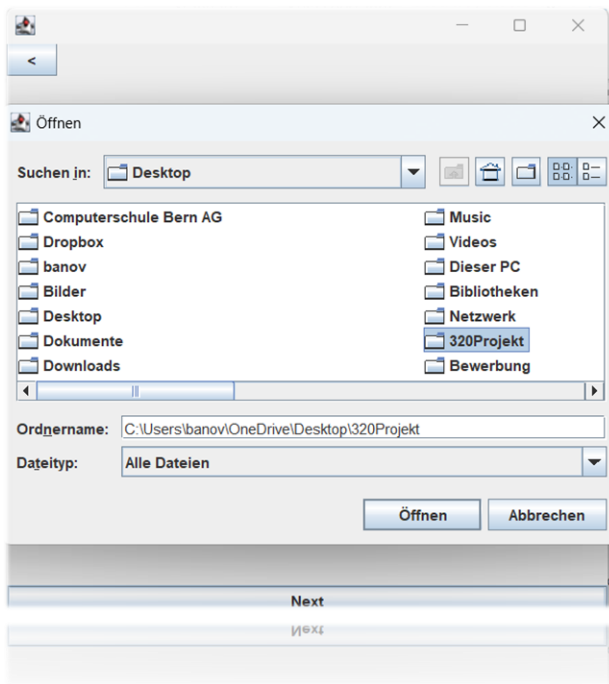


Figure 6

### Ordnerauswahl

Wie erkennbar ist, sollten Sie nun den Ordner auswählen, in dem Sie die Bilder, Texte und Audiodateien gespeichert haben.

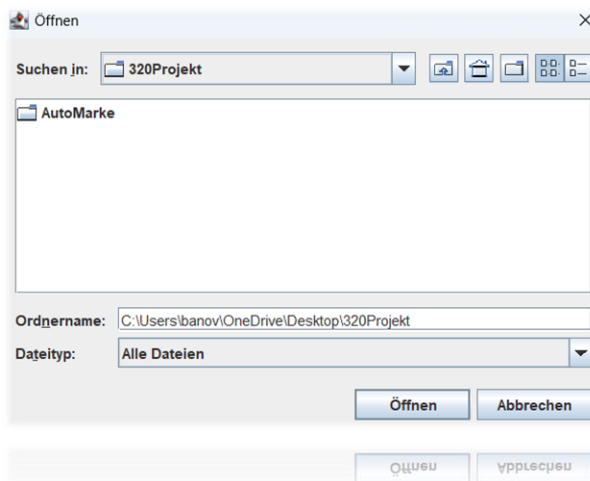


Figure 7

Wie im Beispiel zu sehen ist, navigieren Sie zum Ordner "320Projekt" und anschliessend zum Ordner mit dem Namen "Automarke", der die entsprechenden Inhalte enthält. Nachdem Sie das erledigt haben, klicken Sie bitte auf "Öffnen".

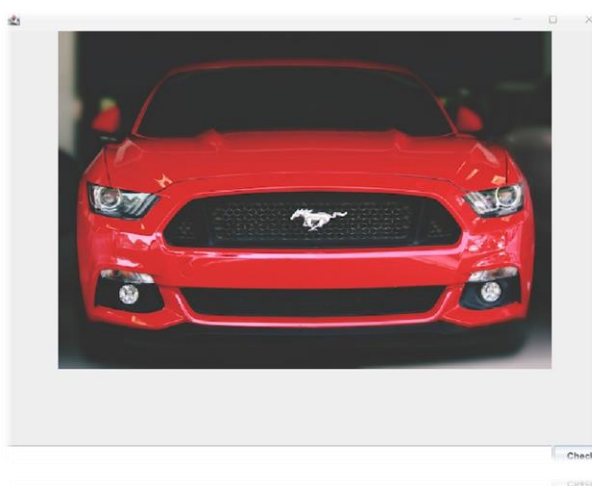


Figure 8

### Lernprozess

Das Spiel hat begonnen. Jetzt müssen Sie erraten, um welches Auto es sich handelt, und den richtigen Namen eingeben. In diesem Beispiel handelt es sich um einen "Ford Mustang". Anschliessend klicken Sie auf "Überprüfen". Wenn die Antwort korrekt ist, wird ein neues Bild, Audio oder Text angezeigt.

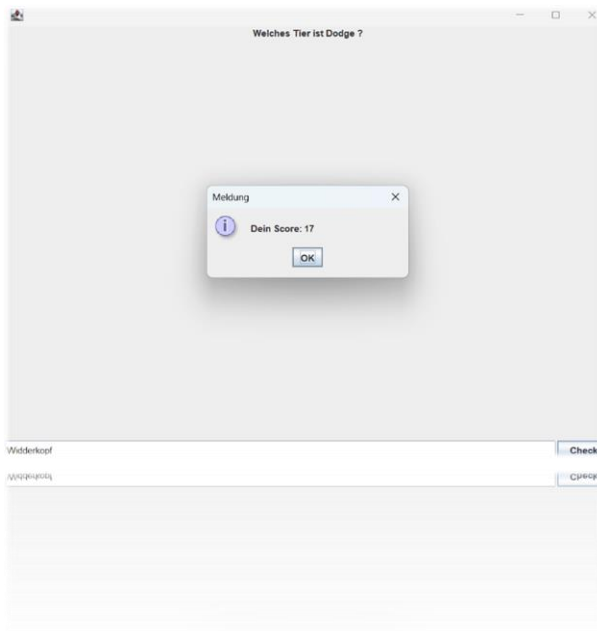


Figure 9

## Score

Nachdem das Programm alle Bilder, Texte oder Audiodateien durchlaufen hat, wird Ihnen eine Punktzahl angezeigt. Dies soll Sie dazu motivieren, sich selbst anzuregen, Aufgaben, die beim letzten Mal länger gedauert haben, beim nächsten Mal schneller zu erledigen. Klicken Sie auf "OK".

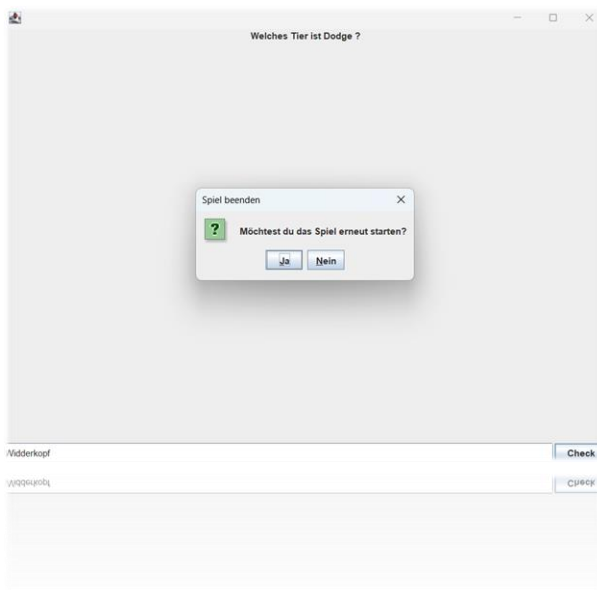


Figure 10

## Ende

Am Ende wird das Programm Sie noch fragen, ob Sie das Spiel erneut starten möchten. Wenn Sie auf "Ja" klicken, beginnt das Spiel von vorne. Wenn Sie hingegen auf "Nein" klicken, wird das Programm das Fenster schließen und beenden.

## Klassendiagramm

### MainWindow:

Die MainWindow-Klasse repräsentiert das Hauptfenster der Anwendung. Es zeigt eine Begrüßungsnachricht und eine Schaltfläche "Start" an, mit der der Benutzer das Spiel initiieren kann. Wenn die Schaltfläche angeklickt wird, öffnet sich das Fenster "StartExit".

- `getInstance()`: Gibt die Singleton-Instanz der MainWindow-Klasse zurück oder erstellt eine, falls keine existiert.
- Konstruktor (`private MainWindow()`): Initialisiert das Hauptfenster, fügt die "Start" Schaltfläche und einen Textbereich hinzu.
- `actionPerformed(ActionEvent event)`: Behandelt Aktionen, insbesondere wenn die "Start" Schaltfläche angeklickt wird, indem es das StartExit-Fenster öffnet.

### StartExit:

Die StartExit-Klasse repräsentiert das Zwischenfenster, das nach dem Klicken auf die "Start" Schaltfläche erscheint. Es gibt eine Erklärung über die Bedeutung von Lernkarten in der digitalen Welt und eine Schaltfläche "Next", um zum Ordner mit den Lernmaterialien zu gelangen.

- `getInstance()`: Gibt die Singleton-Instanz der StartExit-Klasse zurück oder erstellt eine, falls keine existiert.
- Konstruktor (`private StartExit()`): Initialisiert das Zwischenfenster, fügt die "Back" und "Next" Schaltflächen und einen Textbereich hinzu.
- `actionPerformed(ActionEvent event)`: Behandelt Aktionen, insbesondere wenn die "Next" Schaltfläche angeklickt wird, indem es das Ordnerauswahl-Dialogfeld zeigt, oder wenn die "Back" Schaltfläche angeklickt wird, indem es zum Hauptfenster zurückkehrt.

### OrdnerAuswahl:

Die OrdnerAuswahl-Klasse behandelt die Auswahl eines Ordners, in dem die Lernmaterialien gespeichert sind. Es zeigt ein Dialogfeld zur Dateiauswahl an und öffnet danach das QuestionWindow-Fenster mit den Inhalten des ausgewählten Ordners.

- `showFolderSelectionDialog()`: Zeigt ein Dialogfeld zur Auswahl eines Ordners an. Nach der Auswahl wird das QuestionWindow-Fenster mit den Inhalten des Ordners angezeigt.
- `OrdnerAuswahlActionListener`: Eine innere Klasse, die den ActionListener für die "Back" Schaltfläche des Ordnerauswahl-Fensters behandelt.

### QuestionWindow:

Die QuestionWindow-Klasse repräsentiert das Hauptspiel-Fenster, das Bilder, Texte anzeigt und die Benutzerinteraktionen behandelt.

- getInstance(): Gibt die Singleton-Instanz der QuestionWindow-Klasse zurück oder erstellt eine, falls keine existiert.
- Konstruktor (private QuestionWindow()): Initialisiert das Hauptspiel-Fenster, fügt Schaltflächen, Textfeld und Bildanzeige hinzu.
- showImageFromFolder(File folder): Zeigt Bilder und Texte aus dem ausgewählten Ordner an.
- actionPerformed(ActionEvent event): Behandelt Aktionen, insbesondere wenn die "Check" Schaltfläche angeklickt oder eine Antwort im Textfeld eingegeben wird.
- showNextImage(): Zeigt das nächste Bild oder den nächsten Text an.
- showScore(): Zeigt den aktuellen Punktestand an und ermöglicht dem Benutzer, das Spiel neu zu starten oder zu beenden.

### ScoreExit:

Die ScoreExit-Klasse repräsentiert das Fenster, in dem der Benutzer seinen Punktestand anzeigen und Fragen beantworten kann.

- getInstance(): Gibt die Singleton-Instanz der ScoreExit-Klasse zurück oder erstellt eine, falls keine existiert.
- Konstruktor (private ScoreExit()): Initialisiert das Punkte-Anzeige-Fenster, fügt Schaltflächen und Textfeld hinzu.
- actionPerformed(ActionEvent event): Behandelt Aktionen, insbesondere wenn die "Back" Schaltfläche oder die "Check" Schaltfläche angeklickt oder eine Antwort im Textfeld eingegeben wird.

Mein Klassendiagramm bildet eine strukturierte Darstellung der verschiedenen Funktionen und Interaktionen in meiner Anwendung. Es zeigt, wie die unterschiedlichen Klassen zusammenarbeiten, um das Spiel zu ermöglichen, die Lernmaterialien zu verwalten und die Benutzeroberfläche zu steuern.

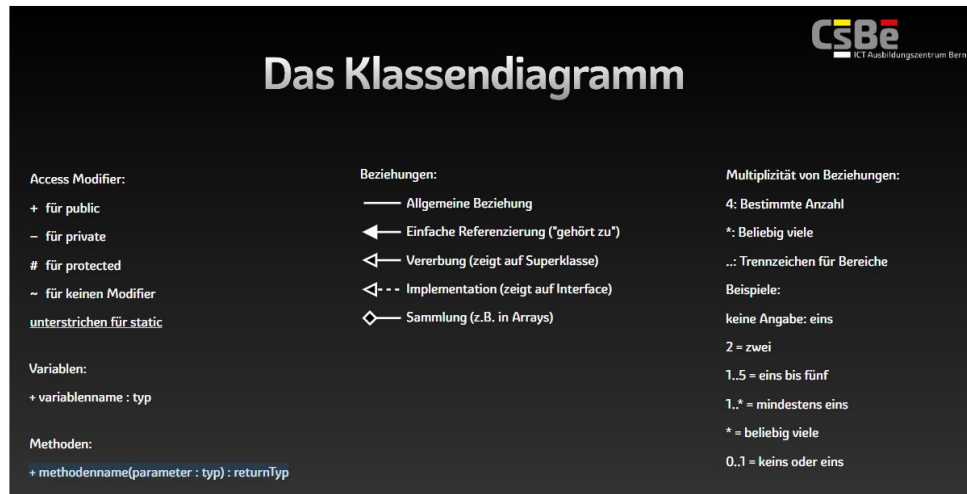


Figure 12

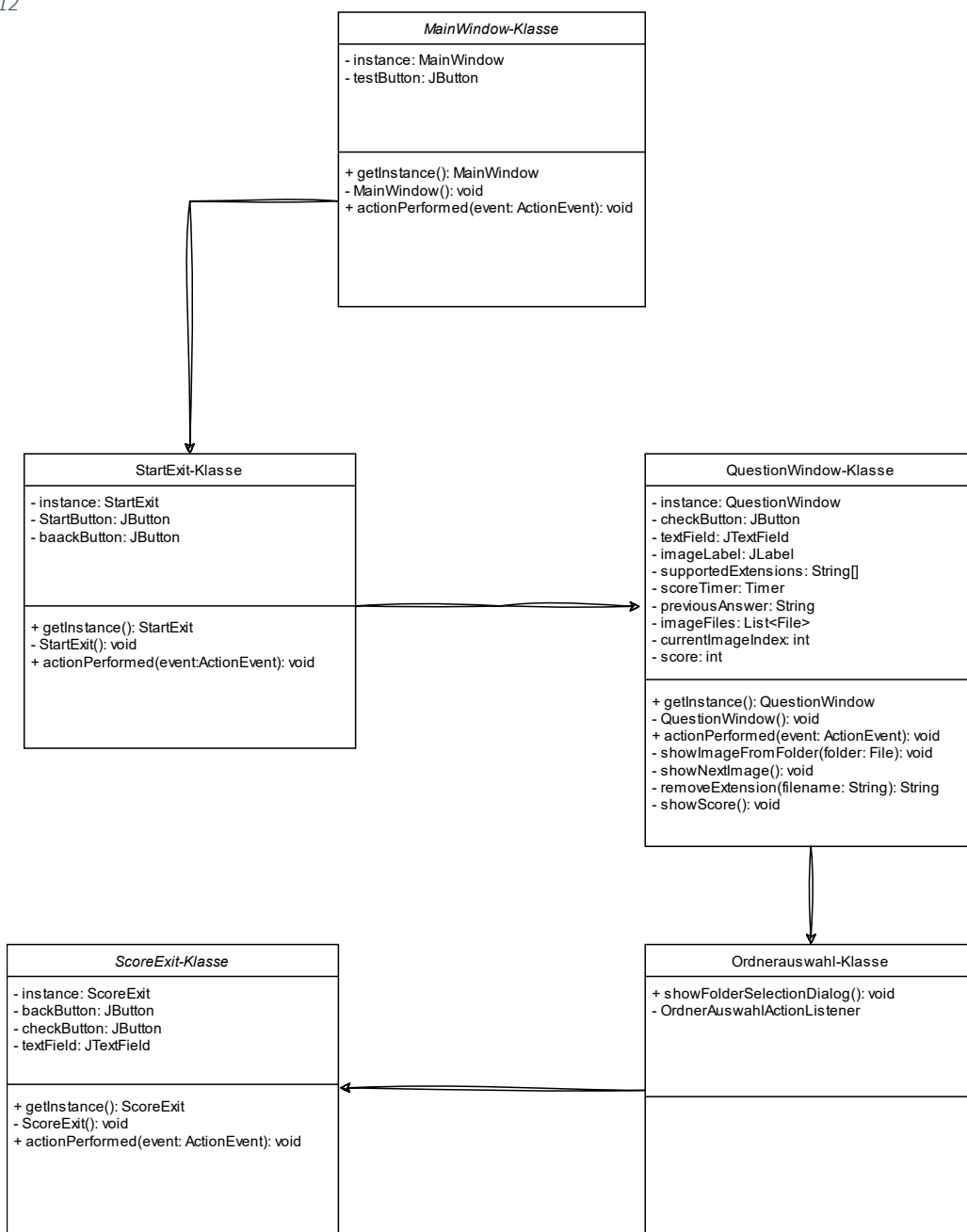


Figure 11

Umsetzung

**Struktur für Flashcards:**

Die Flashcards werden durch ein Datenmodell repräsentiert. Jede Karte enthält eine Frage und eine Antwort.

```
public class Flashcard {  
    private String question;  
    private String answer;  
    public Flashcard(String question, String answer) {  
        this.question = question;  
        this.answer = answer;  
    }  
    public String getQuestion() {  
        return question;  
    }  
    public String getAnswer() {  
        return answer;  
    }  
}
```



**Flashcard-Modell:**

Das Flashcard-Modell verwaltet die Liste der Karten und bietet Methoden zum Hinzufügen, Entfernen und Durchsuchen der Karten.

```
import java.util.ArrayList;
import java.util.List;

public class FlashcardModel {
    private List<Flashcard> flashcards;

    public FlashcardModel() {
        flashcards = new ArrayList<>();
    }

    public void addFlashcard(Flashcard flashcard) {
        flashcards.add(flashcard);
    }

    public List<Flashcard> getAllFlashcards() {
        return flashcards;
    }
}
```

**Benutzeroberfläche:**

Die GUI (grafische Benutzeroberfläche) wird mit Java Swing-Komponenten erstellt. Sie enthält Elemente wie Frage- und Antwortlabels, Schaltflächen zum Anzeigen der Antwort und zum Wechseln zur nächsten Karte.

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class FlashcardGUI extends JFrame {

    private FlashcardModel flashcardModel;

    private JLabel questionLabel;

    private JLabel answerLabel;

    private JButton showAnswerButton;

    private JButton nextCardButton;

    // Konstruktor und ActionListener für Schaltflächen

    private Flashcard getCurrentCard() {

        // Logik, um die aktuelle Karte aus dem Modell zu erhalten

    }

    private void showNextCard() {

        // Logik, um die nächste Karte anzuzeigen

    }

    public static void main(String[] args) {

        FlashcardModel model = new FlashcardModel();

        // Füge Karten zum Modell hinzu

        SwingUtilities.invokeLater(new Runnable() {

            public void run() {

                FlashcardGUI gui = new FlashcardGUI(model);

                gui.setTitle("Flashcard App");

            }

        });

    }

}
```

```
        gui.setSize(400, 300);  
        gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        gui.setVisible(true);  
    }  
    });  
}  
}
```

## Kontrollieren

### Testing

#### Teststrategie

Nachdem sämtliche Features der Anwendung implementiert wurden, wird die Testphase eröffnet. Bei jeder nachfolgenden Änderung, die nach Abschluss der Testphase durchgeführt wird, startet eine neue Testphase, in der das Testprotokoll erneut durchgeführt wird. Es ist wichtig zu beachten, dass keine Version veröffentlicht werden darf, für die das Testprotokoll nicht vollständig abgeschlossen wurde. Die Testphase setzt sich aus zwei Hauptabschnitten zusammen: Zunächst werden die automatisierten Unit-Tests durchgeführt. Anschliessend folgt die Durchführung des funktionalen Tests auf allen Systemen anhand der unten aufgeführten Testfälle.

#### Testumgebung

Da die Anwendung auf Windows 11 laufen muss, wird diese Plattform intensiv getestet. Insbesondere wird die Anwendung auf Windows 11 getestet.

Test auf Windows 11 Wenn bei den einzelnen Tests keine spezifische Fehlermeldung vermerkt ist, wird davon ausgegangen, dass die Tests auf allen getesteten Systemen erfolgreich waren. Falls das tatsächliche Ergebnis auf allen Systemen in gleicher Weise von den Erwartungen abweicht, wird ebenfalls keine Angabe zu den betreffenden Systemen gemacht.

#### Testfall: *Frageanzeige und Bewertung der Benutzerantwort*

**Testziel:** Überprüfen Sie, ob die **QuestionWindow**-Klasse Fragen korrekt anzeigt und Benutzerantworten richtig bewertet.

#### Vorbedingungen:

- I. Die Anwendung ist gestartet.
- II. Ein Ordner mit unterstützten Inhalten (Bilder oder Textdateien) ist ausgewählt.

#### Testablauf:

- I. Starten Sie die Anwendung.
- II. Wählen Sie einen Ordner mit unterstützten Inhalten aus.
- III. Stellen Sie sicher, dass das **QuestionWindow**-Fenster angezeigt wird.
- IV. Überprüfen Sie, ob die erste Frage oder der erste Inhalt aus dem ausgewählten Ordner korrekt angezeigt wird.
- V. Geben Sie eine richtige Antwort in das Antwortfeld ein und drücken Sie die Eingabetaste oder den "Check"-Button.
- VI. Überprüfen Sie, ob die Antwort als korrekt bewertet wird.
- VII. Wiederholen Sie Schritte 4-6 für mehrere Fragen oder Inhalte aus dem Ordner.
- VIII. Geben Sie eine falsche Antwort in das Antwortfeld ein und drücken Sie die Eingabetaste oder den "Check"-Button.
- IX. Überprüfen Sie, ob die Antwort als falsch bewertet wird.
- X. Wiederholen Sie Schritte 4-9, um sowohl korrekte als auch falsche Antworten zu überprüfen.

**Erwartete Ergebnisse:**

- I. Die Fragen oder Inhalte werden korrekt im **QuestionWindow**-Fenster angezeigt.
- II. Korrekte Benutzerantworten werden als richtig bewertet.
- III. Falsche Benutzerantworten werden als falsch bewertet.
- IV. Die Punktzahl erhöht sich korrekt, wenn die Antwort richtig ist.
- V. Die Punktzahl bleibt gleich, wenn die Antwort falsch ist.
- VI. Das **QuestionWindow**-Fenster bleibt sichtbar und aktualisiert sich für jeden neuen Inhalt.

**Nachbedingungen:**

- I. Das **QuestionWindow**-Fenster ist weiterhin sichtbar und zeigt den aktuellen Inhalt oder die Frage an.

**Testergebnis:** Erfolgreich (alle erwarteten Ergebnisse wurden erreicht).

Wenn Fehlermeldungen auftreten oder Änderungen erforderlich sind, um den Code zu verbessern oder anzupassen, gibt es bestimmte Schritte, die Sie unternehmen können

**Bei Fehlermeldungen:**

- I. **Lesen Sie die Fehlermeldung genau:**  
Verstehen Sie, welche Art von Fehler aufgetreten ist und welche Informationen in der Fehlermeldung enthalten sind. Dies wird Ihnen helfen, den Fehlerort und die Ursache besser zu verstehen.
- II. **Überprüfen Sie den betroffenen Code:**  
Gehen Sie zum betroffenen Codeabschnitt und überprüfen Sie ihn auf Tippfehler, syntaktische Fehler oder logische Probleme. Stellen Sie sicher, dass Variablen korrekt initialisiert werden und die Verwendung von Funktionen oder Methoden richtig ist.
- III. **Überprüfen Sie die Dokumentation:**  
Überprüfen Sie die offizielle Dokumentation der verwendeten Bibliotheken, APIs oder Frameworks. Oft finden Sie dort Informationen darüber, wie bestimmte Funktionen verwendet werden sollten oder welche Fehler auftreten können.
- IV. **Debugging-Tools verwenden:**  
Setzen Sie Breakpoints und verwenden Sie Debugging-Tools, um den Programmablauf zu überwachen und den Wert von Variablen zu überprüfen. Dies kann Ihnen helfen, den genauen Fehlerort zu lokalisieren.
- V. **Testen Sie verschiedene Szenarien:**  
Stellen Sie sicher, dass der Fehler in verschiedenen Szenarien reproduziert werden kann. Manchmal tritt ein Fehler nur unter bestimmten Bedingungen auf.

**Bei Änderungen im Code:**

- I. **Verwenden Sie Versionskontrolle:**  
Wenn Sie Änderungen am Code vornehmen, ist es ratsam, eine Versionskontrolle wie Git zu verwenden. Dadurch können Sie Änderungen nachverfolgen, verschiedene Versionen verwalten und im Falle von Fehlern zu einer früheren Version zurückkehren.
- II. **Dokumentieren Sie Änderungen:**  
Kommentieren Sie Änderungen im Code, damit Sie später verstehen können, warum bestimmte Anpassungen vorgenommen wurden.
- III. **Überprüfen Sie die Auswirkungen:**  
Testen Sie den Code nach den Änderungen gründlich, um sicherzustellen, dass keine neuen Fehler eingeführt wurden und dass bestehende Funktionen weiterhin ordnungsgemäss arbeiten.
- IV. **Beachten:**  
Stellen Sie sicher, dass die Änderungen den bewährten Methoden und Entwicklungsstandards entsprechen. Dies hilft, die Codequalität aufrechtzuerhalten.
- V. **Zusammen Programmieren:**  
Arbeiten Sie bei komplexen Änderungen oder Fehlerbehebungen mit einem Kollegen zusammen. Zwei Köpfe denken oft besser als einer, und es können Fehler leichter erkannt werden.
- VI. **Behalten Sie den Überblick:**  
Wenn Sie umfangreiche Änderungen vornehmen, behalten Sie den Überblick über den gesamten Code und stellen Sie sicher, dass alle Abhängigkeiten und Zusammenhänge berücksichtigt werden.

## Unit-Testing

Hier ist ein Beispiel für einen Unit-Test, der überprüft, wie viele Dateien im Ordner vorhanden sind, wenn die Methode `showImageFromFolder` aufgerufen wird. Dieser Test stellt sicher, dass die Methode korrekt funktioniert und die erwartete Anzahl von Dateien zurückgibt. Wie auf diesem Bild zu sehen ist, wird die Anzahl korrekt angezeigt und durch einen grünen Balken markiert.



Figure 13

Wenn die Anzahl falsch ist, wird der Balken unten im Bild in Rot angezeigt.

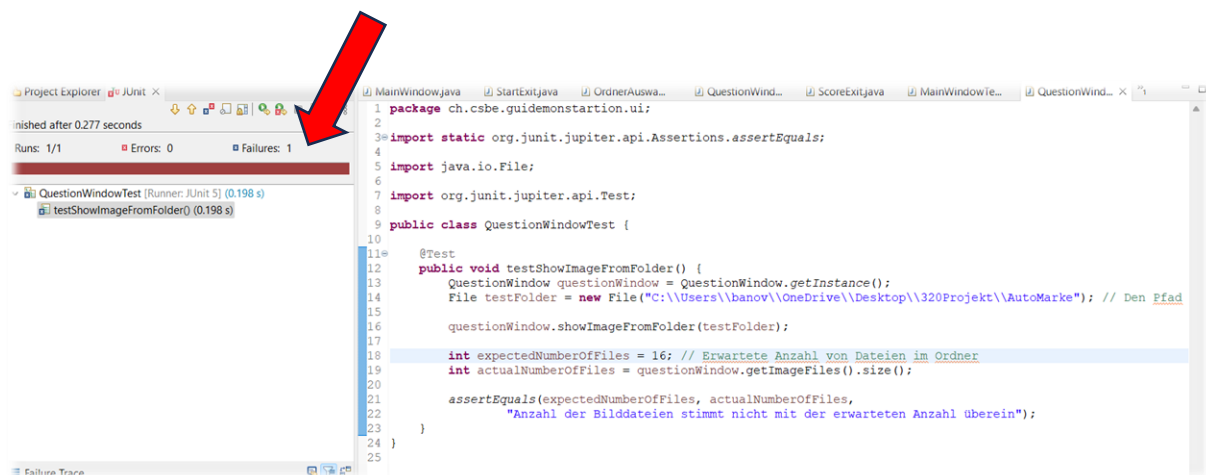


Figure 14

## Arbeitsjournal

Tag 1	
Tätigkeit	Arbeitszeit (h)
Dokumentation erstellen	0.75
Projektorganisation beschreiben	0.25
Ausgangslage beschreiben	0.25
Projektziele beschreiben	0.5
Anforderung definieren	0.5
Rollen und User Stories definieren	0.5
Funktionen planen	1.5
GUI-Layout entwerfen	1.5
UI-Interaktionen planen	0.5
<b>Reflektion</b> Heute habe ich mit meinem Projekt angefangen. Während ich die Funktionen beschrieben habe, habe ich mehr Zeit gebraucht als gedacht, weil ich lange überlegen musste, ob alles korrekt ist. Bei der Planung meines Projekts bin ich ohne grössere Probleme vorangekommen. Als nächstes werde ich die Klassen und die dazugehörigen Pakete entwerfen.	

Tabelle 4

Tag 2	
Tätigkeit	Arbeitszeit (h)
Klassendiagramm erstellt	3.0
Packages erstellt	2.0
<b>Reflektion</b> Heute habe ich zuerst ein Klassendiagramm erstellt und anschliessend die Pakete angelegt. Währenddessen habe ich eine Weile darüber nachgedacht, ob das, was ich tue, richtig ist, und dabei habe ich die meiste Zeit verloren.	

Tabelle 5

Tag 3	
Tätigkeit	Arbeitszeit (h)
MainWindow. klasse erstellt und Code ausgeschrieben	3.0
StartExit klasse erstellt und Code ausgeschrieben	2.5
<b>Reflexion:</b> Heute habe ich gemeinsam mit Mitric die Klassen erstellt und sie mit Code gefüllt. Darüber hinaus habe ich die Schaltflächen erstellt. Ich stiess auf Schwierigkeiten beim Back-Button, da der Next-Button nicht ordnungsgemäss funktionierte. Dank einiger Recherche im Internet konnte ich das Problem jedoch lösen.	

Tabelle 6



Tag 4	
Tätigkeit	Arbeitszeit (h)
Ordnerauswahl klasse erstellt und Code ausgeschrieben	4.0
QuestionWindow klasse erstellt und Code ausgeschrieben	2.5
Reflektion Heute habe ich die Klassen "OrdnerAuswahl" und "QuestionWindow" mit Code ausgestattet. Dabei stiess ich auf Schwierigkeiten bei der Verknüpfung zum Anzeigen von Bildern und Textdateien. Ich konnte erfolgreich die Klasse "OrdnerAuswahl" mit der Klasse "QuestionWindow" verknüpfen. Allerdings konnte ich das Problem beim Anzeigen der Bilder noch nicht lösen. Ich werde weiter daran arbeiten, um eine Lösung zu finden.	

Tabelle 7

Tag 5	
Tätigkeit	Arbeitszeit (h)
QuestionWindow Problem Bild anzeigen lassen lösen.	4.0
QuestionWindow Problem Text anzeigen zu lassen.	6.0
Reflexion: Heute konnte ich erfolgreich die Bilder anzeigen lassen, jedoch habe ich weiterhin Schwierigkeiten damit, die Fragen aus den Textdateien anzuzeigen. Trotz vieler Versuche konnte ich dieses Problem nicht lösen und habe mich entschlossen, vorerst damit aufzuhören. Ich werde mich weiterhin bemühen, eine Lösung zu finden.	

Tabelle 8

Tag 6	
Tätigkeit	Arbeitszeit (h)
QuestionWindow Problem Text anzeigen zu lassen.	6.0
ScoreExit klasse erstellt und Code ausgeschrieben.	2.5
Reflektion Heute habe ich erneut versucht, die Textdateien mit den Fragen anzuzeigen. Trotz stundenlanger Bemühungen war ich nicht erfolgreich. Frustriert von diesem Problem habe ich mich dazu entschieden, vorerst eine Pause einzulegen und stattdessen mit der ScoreExit-Klasse zu arbeiten. Hierbei hatte ich mehr Glück und konnte sie erfolgreich mit Code ausstatten. Dies hat mir etwas mehr Zuversicht gegeben, auch wenn das Problem mit den Textdateien weiterhin ungelöst bleibt.	

Tabelle 9

Tag 7	
Tätigkeit	Arbeitszeit (h)
QuestionWindow Problem Text anzeigen zu lassen.	4.0
Dokumentation	2.0
Reflektion Heute hatten wir Online-Unterricht, und Herr Sollberger hat mir wichtige Informationen gegeben, die mir geholfen haben, im Code Fortschritte zu erzielen. Danach hatten wir ein Teammeeting mit Herrn Leimbacher und Mitric. Auch sie haben mir bei meinem Code etwas Unterstützung angeboten. Zum Abschluss habe ich noch an der Dokumentation gearbeitet.	

Tabelle 10

Tag 8	
Tätigkeit	Arbeitszeit (h)
QuestionWindow Problem Text anzeigen zu lassen.	3.0
An der Dokumentation gearbeitet	3.0
Reflektion Heute ist es mir gelungen, den angezeigten Text aus der TXT-Datei korrekt anzuzeigen. Zuvor habe ich meine Zeit damit verbracht, an der Dokumentation zu arbeiten.	

Tabelle 11

Tag 9	
Tätigkeit	Arbeitszeit (h)
Test durchgeführt und Unit- Tests	2.0
An der Dokumentation gearbeitet	6.0
Reflektion Heute habe ich zwei Stunden damit verbracht, Tests durchzuführen, darunter auch Unit-Tests. Anschliessend habe ich weitere sechs Stunden an der Dokumentation gearbeitet und dabei Verbesserungen und Erweiterungen vorgenommen.	

Tabelle 12

## Auswerten

### Fazit

Digitale Karteikarten sind für mich eine äusserst praktische Ergänzung meiner Lernroutine. Ihre elektronische Natur ermöglicht es mir, meine Karten über verschiedene Geräte hinweg zu synchronisieren, was flexibles Lernen fördert. Die Möglichkeit, multimediale Elemente wie Bilder und Audio einzufügen, hilft mir dabei, komplexere Konzepte besser zu verstehen und zu behalten. Zudem finde ich es bequem, automatische Erinnerungen für das Wiederholen einzustellen, was mir hilft, den Lernstoff effizienter zu verwalten. Der Zugang zu Online-Plattformen für das Teilen und Herunterladen von Karten erleichtert auch das Entdecken neuer Perspektiven und Ansätze beim Lernen. Insgesamt haben digitale Karteikarten definitiv meine Lernprozesse bereichert.

### Persönliches Fazit:

Das Arbeiten mit Karteikarten, sei es auf Papier oder digital, hat mein Lernen auf viele Arten bereichert. Sie ermöglichen mir ein aktives, selbstgesteuertes Lernen, bei dem ich mich intensiv mit den Inhalten auseinandersetze. Die regelmässigen Wiederholungen, sei es durch traditionelles Karteikartenlernen oder digitale Tools, haben mir geholfen, das Gelernte besser zu behalten und in meiner Erinnerung zu verankern. Die Möglichkeit, meine eigenen Karten zu erstellen, gibt mir die Freiheit, den Lernprozess an meinen individuellen Bedürfnissen auszurichten. Es ist beeindruckend, wie eine einfache Methode wie Karteikarten meine Fähigkeit, Wissen zu erwerben und zu behalten, erheblich verbessern kann.

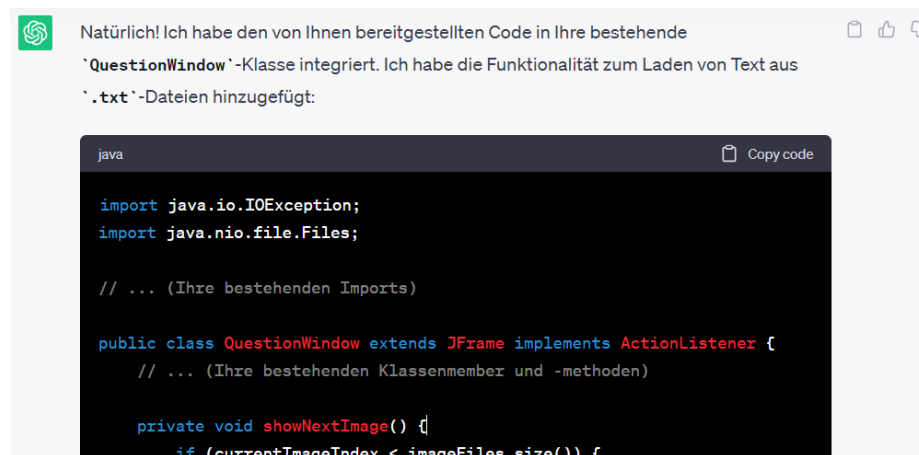


Figure 15

## Quellen

- Manuel Sollberger
- Unterricht von Manuel Sollberger
- Unterlagen von Manuel Sollberger
- Schulkameraden
- <https://de.wikipedia.org/wiki/Javadoc#:~:text=Javadoc%20ist%20ein%20Software%2DDokumentationswerkzeug,Bestandteil%20des%20Java%20Development%20Kits.>  
06.07.2023
- <https://www.futura-sciences.com/de/wp-content/uploads/2022/07/definition-von-objektorientierte-programmierung.jpg>  
06.07.2023
- <https://www.youtube.com/watch?v=4sVJuH0NdSw&t=311s>  
20.07.2023
- <https://www.java-tutorial.org/>  
22.07.2023
- Ich habe einen Code erhalten, aber die hat mir nicht viel gebracht weil ich die code noch selber abändern durfte.  
18.09.2023

Figure 16



- <https://de.wikipedia.org/wiki/Javadoc>  
19.08.2023
- <https://entwickler.de/java/neues-von-junit>  
19.08.2023
- <https://kompf.de/java/junit.html>  
19.08.2023
- <https://entwickler.de/programmierung/ganz-unkompliziert-so-werden-code-dokumentationen-richtig-geschrieben>  
19.08.2023
- <https://sozialministerium.baden-wuerttemberg.de/de/gesundheitspflege/gesundheitschutz/infektionsschutz-hygiene/informationen-zu-coronavirus/informationen-in-mehreren-sprachen/>  
19.08.2023
- <https://www.testbirds.com/de/>  
19.08.2023
- <https://www.freecodecamp.org/news/java-unit-testing/>  
19.08.2023
- <https://www.vogella.com/tutorials/JUnit/article.html>  
19.08.2023
- [https://fernstudium.s3.eu-central-1.amazonaws.com/monthly\\_2018\\_10/stempel\\_fazit\\_gruen-320x145.jpg.928fb2b14fcea611e3c76fe780bba079.jpg](https://fernstudium.s3.eu-central-1.amazonaws.com/monthly_2018_10/stempel_fazit_gruen-320x145.jpg.928fb2b14fcea611e3c76fe780bba079.jpg)

- 19.08.2023
- <https://www.kompf.de/java/junit.html>  
17.08.2023

## Abbildungsverzeichniss

Figure 1 .....	0
Figure 2 .....	4
Figure 3 .....	7
Figure 4 .....	9
Figure 5 .....	9
Figure 6 .....	10
Figure 7 .....	10
Figure 8 .....	10
Figure 9 .....	11
Figure 10 .....	11
Figure 11 .....	14
Figure 12 .....	14
Figure 13 .....	22
Figure 14 .....	22

## TabellenVerzeichniss

Tabelle 1 .....	6
Tabelle 2 .....	6
Tabelle 3 .....	8
Tabelle 4 .....	23
Tabelle 5 .....	23
Tabelle 6 .....	23
Tabelle 7 .....	24
Tabelle 8 .....	24
Tabelle 9 .....	24
Tabelle 10 .....	24
Tabelle 11 .....	25
Tabelle 12 .....	25