

Meilenstein 3 – Programmierprojekt

Aufgabenstellung

Der letzte Meilenstein des OOP-Praktikums ist als offenes Programmierprojekt ausgestaltet, das Sie wieder in (den schon bestehenden) Gruppen von 2 bis 4 Personen bearbeiten sollen. Für das Projekt gelten folgende Regeln:

Anwendung der erlernten Konzepte

In Ihrem Projekt sollen Sie die bislang in der OOP-Vorlesung erlernten Konzepte umsetzen. D.h., Ihr Code muss folgende Anforderungen erfüllen:

- Sie setzen **Vererbung** ein, indem Sie eigene Klassenhierarchien implementieren. Es muss also eine selbst implementierte Klasse mit mehr als einer abgeleiteten Klasse geben.
- Sie verwenden **Interfaces**, d.h. mindestens einer Ihrer Klassen implementiert ein *notwendiges* Interface. Sie dürfen gerne auch eigene Interfaces definieren, oder vorhandene aus dem .NET-Standard implementieren.
- Sie arbeiten mit **Exceptions**, um Fehler/Ausnahmefälle kontrolliert zu behandeln.
- Sie verwenden mindestens eine **generische, dynamische Datenstruktur**, um die Daten Ihrer Applikation zu verwalten.
- Sie nutzen **Delegates**, z.B. um ein User Interface im Sinne der *Ereignis-basierten Programmierung* umzusetzen.

Verwendung von GitLab

Versionsverwaltungen sind ein zentrales Tool in der Softwareentwicklung für die Arbeit im Team. Spätestens mit diesem Programmierprojekt sollen Sie die ersten Erfahrungen in der Verwendung von Git sammeln. Daraus ergeben sich folgende Anforderungen an Ihr Projekt:

- Sie müssen ein gemeinsames Repository im GitLab der Fakultät Informatik aufsetzen und im Rahmen Entwicklung in diesem Projekt nutzen.
- Alle Projektbeteiligten committen regelmäßig ihren Quellcode in das GitLab-Repository, sodass die Beiträge aller sichtbar werden.
- Sie laden den Dozenten, Prof. Scholz, als Gast in Ihr GitLab-Projekt ein. Mit dem entsprechenden Eintrag im Moodle, dass Sie Ihr Projekt abgeschlossen, gilt es dann auch als abgegeben. Abgabefrist ist der **17. Juni 2025** um 23:59 Uhr.

Umfang des Projekts

Ihr Projekt muss eine sinnvolle Aufgabe lösen. Dazu können Sie eines der unten genannten Beispielthemen verwenden und implementieren, oder sich ein eigenes Projektziel überlegen.

Wichtig: Sie müssen Ihr Projekt über Moodle **anmelden** und sich Ihr Thema bestätigen lassen. Das gilt *insbesondere*, wenn Sie sich ein eigenes Thema überlegen. Dann geben Sie bitte, analog zu den Beispielprojekten unten, die Features an, die Sie umsetzen wollen.

Die Anforderungen an die einzusetzenden Konzepte und die Liste der Features bestimmen den ungefähren Umfang des Projekts. Sollte Ihr Projekt zu ehrgeizig oder zu klein dimensioniert sein, bekommen Sie eine entsprechende Rückmeldung Ihres Dozenten.

Präsentation und Bewertung

In den Vorlesungen am **23.6.2025** haben Sie die Gelegenheit, Ihr Projekt in einer kurzen Live-Demo als Gruppe vorzustellen. Die Demo sollte nicht länger als drei Minuten dauern. Anschließend werden Sie noch zu Ihrer Implementierung befragt, um sicherzustellen, dass Sie alle Konzepte verstanden haben. Dazu halten Sie bitte den Quellcode auf Ihrem Laptop bereit.

Wichtig: Sie erhalten einen Präsentationsslot in einer der beiden Vorlesungen am 23.6., zu dem Sie auch als gesamte Gruppe anwesend sein müssen. Der Besuch der jeweils anderen Vorlesung ist dann nicht notwendig. Sollten Sie Ihren Slot tauschen wollen, melden Sie sich bitte beim Dozenten.

Ausweichtermin für besondere Fälle, oder falls die Zeit am 23.6. nicht ausreichen sollte, ist der 30.6.2025.

Bedingungen für das Bestehen des dritten Meilensteins sind:

- Nachweisliche, nicht-triviale Mitarbeit über Commits im GitLab-Projekt.
- Anwesenheit bei der Projektpräsentation.
- Korrekte Beantwortung der Fragen im Anschluss an die Projektpräsentation.

Beispielprojekte

Es ist ausdrücklich erwünscht, dass Sie sich eigene Ideen für Ihre Projekte überlegen. Die folgenden Beispiele sollen Ihnen als Inspiration dienen. Sollten Sie aber keine eigenen Ideen haben, können Sie auch eines dieser Projekte umsetzen.

Rezeptverwaltung

Eingabe von eigenen Rezepten und Suche nach Rezepten nach Zutat. Das Verwaltungssystem soll folgende Anforderungen erfüllen:

- Es existiert ein grafisches Benutzerinterface (z.B. über Windows Presentation Forms).
- Über eine Maske werden Rezepte eingegeben. Ein Rezept besteht aus einem Namen, einer Liste von Zutaten und zugehörigen Mengenangaben, sowie einer Beschreibung.
- Die Rezepte werden in einer JSON-Datei persistent gespeichert und beim Öffnen des Programms wieder geladen.
- Rezepte können über eine Liste der verfügbaren Rezepte ausgewählt oder über eine Suchfunktion (nach Name oder Zutat) gefunden werden.
- Rezepte werden einzeln angezeigt, nachdem sie ausgewählt wurden.

Lern-Dashboard

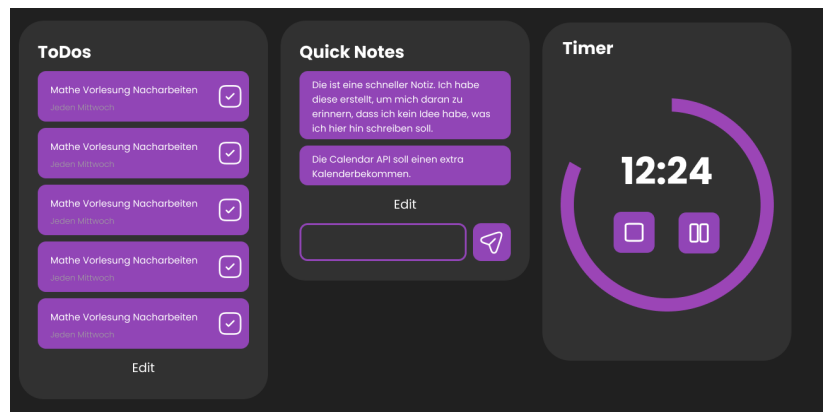
Ein Programm, das beim Lernen helfen und den Prozess strukturieren soll. Dazu kann man Termine, Aufgaben und Notizen anlegen und verwalten. Anforderungen:

Bedienoberfläche

- Dashboard, Navigationsleiste und Detailanzeigen
- Komponentenweise Aufteilung
- Gestaltung mit WPF

Datenerfassung und Aufbereitung

- Aufgabenverwaltung: Erstellen von (wiederholenden) ToDos
- Terminverwaltung mit der Google Calendar API (optional)
- Timer für die Lernzeit bis zur nächsten Pause
- Persistente Speicherung der Daten in einer Datenbank oder einer JSON-Datei



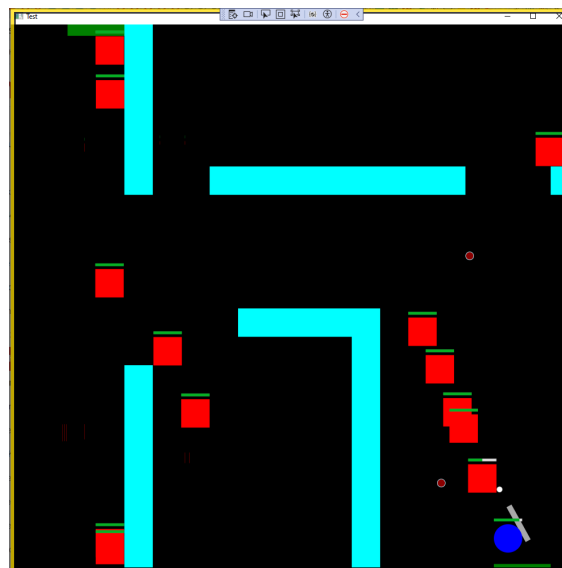
Beispiel Lern-Dashboard (Bild: Stähler, Luttkus, Horn, Schmuck, Schubert)

Top-Down-Shooter

Ein actiongeladener, raumbasierter Shooter, bei dem der Spieler gegen eine Horde zulaufender Gegner kämpfen muss.

Anforderungen:

- Die Applikation verwendet Windows Forms/MonoGame/Unity, um ein grafisches UI anzuzeigen.
- Der Spieler kann über die Tastatur gesteuert werden und schießen. Die Gegner bewegen sich automatisch.
- Spieler und Gegner besitzen Hitpoints, die angezeigt werden.
- Ein Level bzw. das Spiel endet, wenn entweder alle Gegner abgeschossen wurden oder der Spieler keine Hitpoints mehr hat.



Beispiel Top-Down-Shooter von 2024 (Design: Arpert, Djanbazi, Ganzleben, Hager, Viertel)