

## Aufgabe 1

In Aufgabe A6 auf Aufgabenblatt Nr. 1 wurde bereits eine mobile App für die Organisation von Mitfahrgelegenheiten zur FH Aachen besprochen, die Sie auf den Markt bringen wollen. Beschreiben Sie, welche Funktionalitäten ein Minimum Viable Product (MVP) für diese Anwendung enthalten könnte. Geben Sie außerdem zwei Ausbaustufen für die Zukunft an, wenn sich der MVP auf dem Markt bewährt hat.

### Lösung 1

Bei einem **Minimum Viable Product (MVP)** müssen die Grundfunktionalitäten so erfüllt werden, dass die Software benutzbar ist und zur Weiternutzung anregt.

Im Falle einer Mitfahrgelegenheiten-App bedeutet dies, dass Nutzende sich über Fahrten informieren können und einer Fahrt zusagen können. Wichtig ist für das MVM auch schon ein Authentifizierungssystem, da die Datensicherheit zu jeder Nutzungszeit gewährleistet sein muss. - Kurz gesagt, müssen zumindest Mitfahrgelegenheiten zustande kommen können.

In der **ersten Ausbaustufe** könnte die App dann um eine Option zur Bewertung der Fahrt erweitert werden oder es könnte die Möglichkeit geboten werden, mit verschiedenen Filterkriterien nach Fahrten zu suchen.

In einer **zweiten Ausbaustufe** könnte eine Bezahlungsfunktion eingerichtet werden oder etwa ein automatisiertes Matchmaking von Personen mit ähnlichem Reiseziel implementiert werden. Auch Benachrichtigungen, wenn Fahrten angeboten werden, nach denen man zuvor gesucht hat, könnte Teil einer zweiten Ausbaustufe sein.

## Aufgabe 2

Modellieren Sie den folgenden Prozess für Urlaubsanträge als UML-Aktivitätsdiagramm:

- Die Antragstellenden beantragen den Urlaub durch Angabe des Zeitraums und Benennung einer stellvertretenden Person. Der Antrag ist dann im Zustand Neu.
- Wenn die Anzahl der beantragten Urlaubstage das Urlaubskontingent der antragstellenden Person für das entsprechende Kalenderjahr überschreitet, dann wird der Antrag sofort abgelehnt.
- Ansonsten müssen die genannte stellvertretende Person und unabhängig davon die abteilungsleitende Person den Antrag bestätigen. Die Reihenfolge ist dabei unerheblich.
  - Wenn einer der beiden Personen den Antrag nicht bestätigt, dann wird dieser sofort abgelehnt.

- Erst wenn beide Personen bestätigt haben, ändert sich der Zustand des Antrags auf Genehmigt.
- Sollte eine Person nicht innerhalb von fünf Arbeitstagen reagieren, dann gilt der Antrag automatisch als bestätigt.
- Abgelehnte oder genehmigte Anträge werden immer archiviert. Danach endet der Prozess.

## Lösung 2

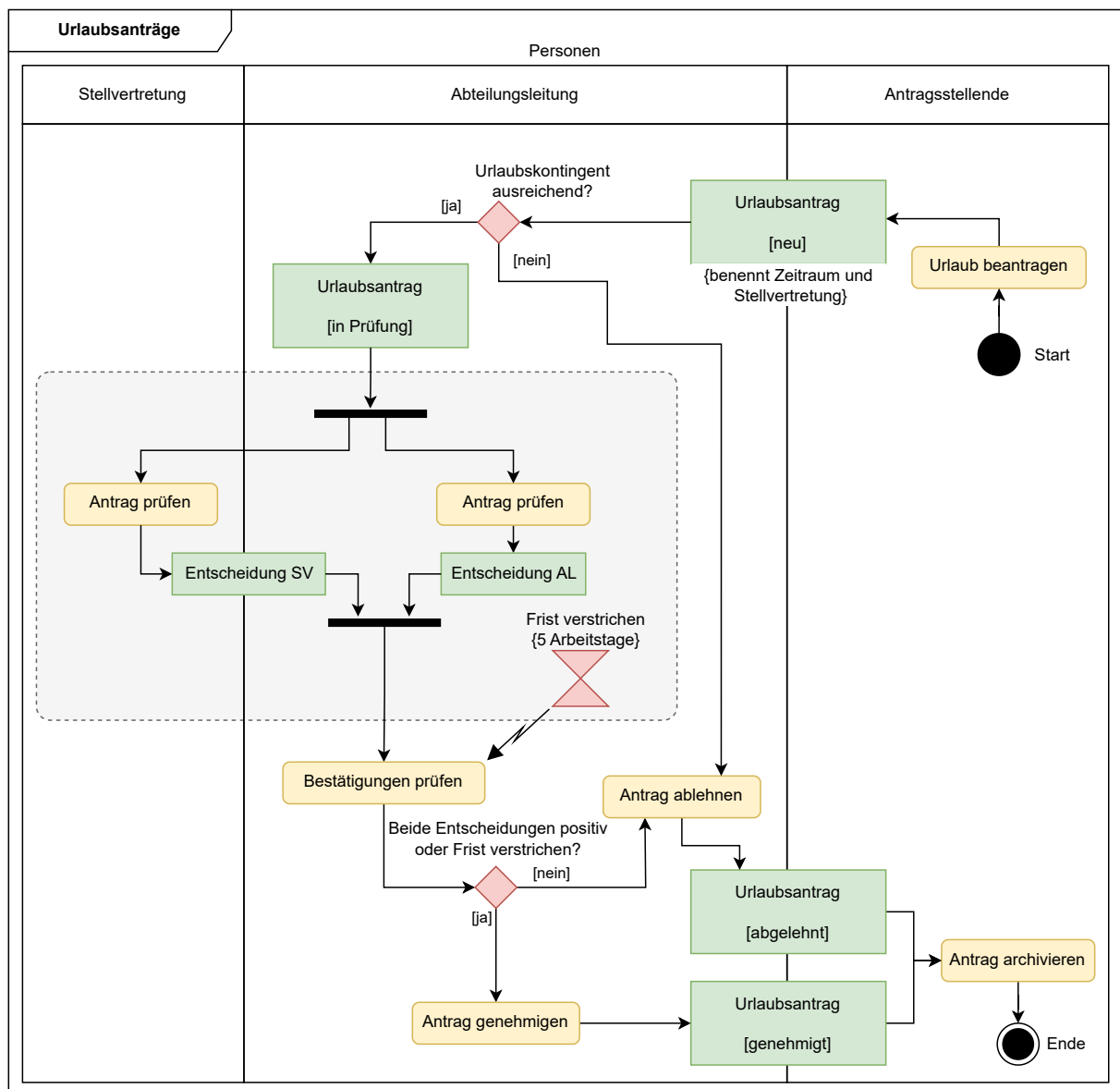


Abbildung 1: Lösung der Aufgabe 2

## Aufgabe 3

In Aufgabe A5 auf Aufgabenblatt Nr. 1 wurden bereits Anwendungsfälle für einen Fahrkartenautomat einer regionalen Privatbahn besprochen. Der Ablauf des Anwendungsfalls »Fahrkartenkauf« soll nun präzisiert werden. Die zugehörigen Aktionen lauten:

- Verbindung suchen und auswählen
- Passende Fahrkarte auswählen
- Optional: Verbindungsdetails ausdrucken lassen
- Geld einwerfen
- Fahrkarte entnehmen
- Ggf. Rückgeld entnehmen
- Ggf. Verbindungsdetails entnehmen

Erst wenn ausreichend Geld eingeworfen wurde, wird die Fahrkarte gedruckt. Bis dahin kann der Fahrkartenkauf jederzeit durch den Benutzer abgebrochen werden. Sollte in diesem Fall bereits Geld eingeworfen worden sein, dann wird es wieder ausgegeben. Der Fahrkartenautomat nimmt nur Bargeld an. Modellieren Sie den Fahrkartenkauf als UML-Aktivitätsdiagramm aus Sicht der Nutzenden. Modellieren Sie dabei auch die aus Sicht der Nutzenden relevanten Objekte und Objektflüsse. *Hinweis: den Fall, dass der Fahrkartenautomat eventuell nicht genügend Rückgeld ausgeben kann, können Sie bei der Modellierung ignorieren.*

## Lösung 3

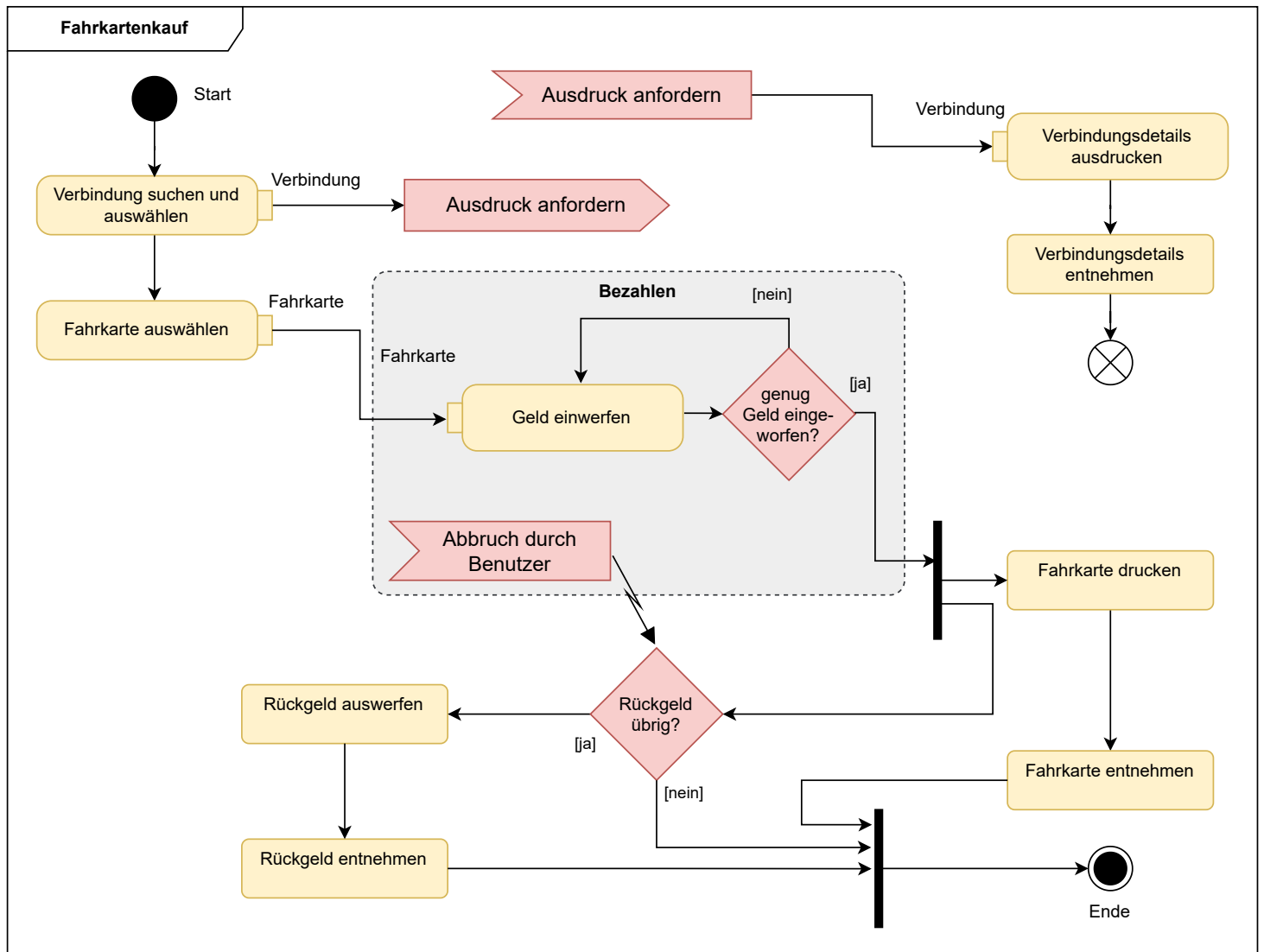


Abbildung 2: Lösung der Aufgabe 3

## Aufgabe 4

Modellieren Sie den klassischen Euklidischen Algorithmus in seiner iterativen Form formal korrekt als Aktivität eines UML-Aktivitätsdiagramms, siehe [Wikipedia: Euklidischer Algorithmus](#). Vergessen Sie dabei nicht, auch die Ein- und Ausgabeparameter und die Datentypen anzugeben.

## Lösung 4

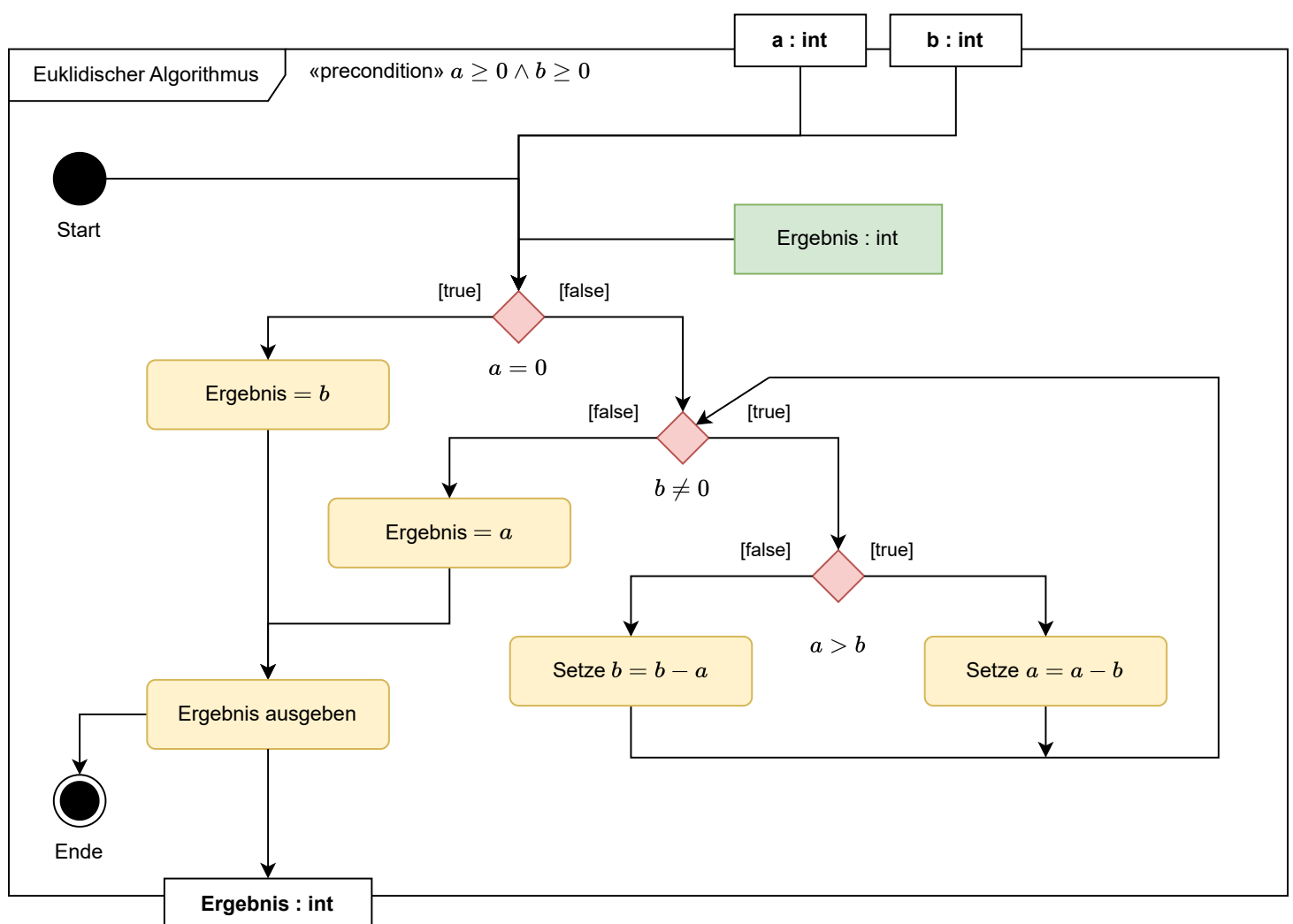


Abbildung 3: Euklidischer Algorithmus