

## # Programmierung Prüfung

### ## Anweisungen

- Beantworte alle Fragen in diese Markdown Datei.
- Wenn eine Frage mit ein Bild beantwortet werden soll, lade das Bild in diesen Repository hinzu und schreibe eine Referenz dazu in diese Datei. Zum referenzieren des Bildes benötigen Sie dieser Befehl: `![Alt text](image_filename.png)`.
- Beispielsbild:  
`![Schöner Strand](sampleImage.jpg)`

---

### **## Aufgabe 1: Code Verständnis**

**\*\*Was macht der Folgende Code?\*\***

```
```csharp
static List<int> VerarbeiteZahlen(List<int> liste)
{
    List<int> neueListe = new List<int>();

    foreach (int zahl in liste)
    {
        if (zahl % 2 == 0 && zahl > 5)
        {
            neueListe.Add(zahl * 3);
        }
    }

    neueListe.Sort();
    return neueListe;
}
```

**\*\*Antwort:\*\***

<!-- Schreiben Sie Ihre Antwort nach diesen Grünen Teil und oberhalb der drei Striche. -->

Der Code stellt eine Funktion stellt eine Liste von Zahlen dar, die gefiltert werden. Fügen wir Zahlen in diese Liste ein, so werden alle Zahlen die größer als 5 und gleichzeitig gerade sind. "zahl % 2 == 0" beschreibt, dass beim teilen durch 2, kein rest übrig bleibt. Alle zahlen die so gefiltert werden, werden dann mit 3 multipliziert und der Liste wieder hinzugefügt.

---

### **## Aufgabe 2: Methoden Schreiben**

**\*\*Schreibe eine Funktion in C# der einen String rückwärts auf die Konsole ausgeben würde, ohne eingebaute reverse funktionen zu verwenden.\*\***

```
```csharp
static void StringReversal(string text)
{
    conwole.write("Schreibe etwas:");
    string vorwärts = console.readline();
    textRückwärts(vorwärts);

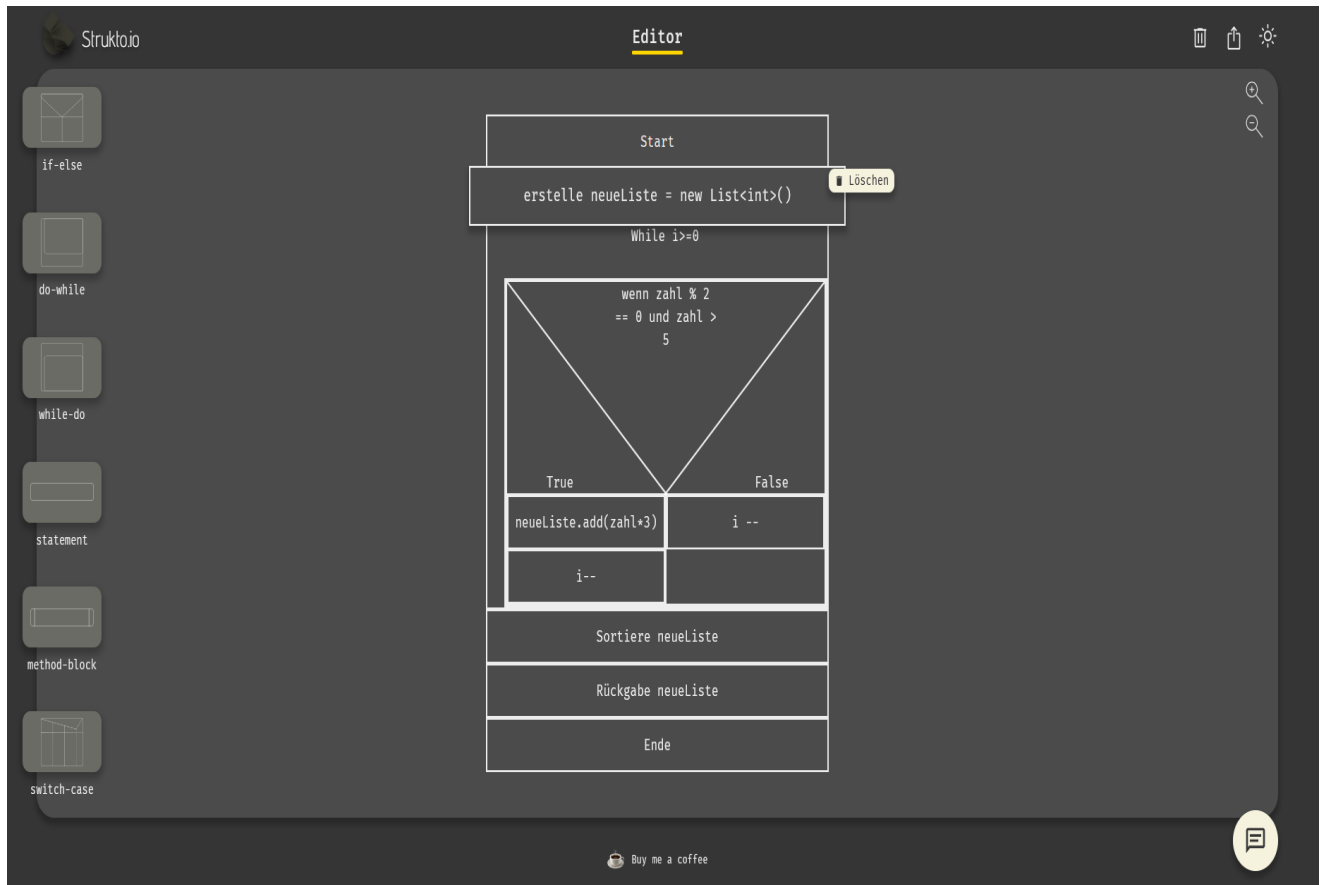
    static void textRückwärts(string wort)
    {
        for (l = wort.length -1; i>=0;i--)
        {
            console.write(text[i]);
        }
        console.writeline();
    }

}
...
---
```

### ## Aufgabe 3: Struktogramm Erstellen

**\*\*Erstelle einen Struktogramm, dass die Aufgabe 1 repräsentiert.\*\***

Hier das Bild von deinem Struktogramm hochladen:



## ## Aufgabe 4: UML Klassendiagramm Erstellen

**\*\*Beispielszenario: Ticket-Management-System\*\***

Übersicht:

Ein Unternehmen nutzt ein Ticket-Management-System, um Kundenprobleme zu verfolgen und

zu lösen. Kunden können Tickets für technischen Support, Rechnungsanfragen oder allgemeine Unterstützung erstellen. Support-Mitarbeiter bearbeiten diese Tickets, während Manager den Prozess überwachen.

**\*\*Entitäten & Ihre Rollen:\*\***

Kunde (Customer)

Ein Kunde erstellt und betrachtet Tickets.

Kunden erhalten Updates über den Status ihrer Tickets.

Support-Mitarbeiter (Support Agent)

Ein Support-Mitarbeiter wird einem Ticket zugewiesen.

Support-Mitarbeiter aktualisieren den Ticketstatus und geben Lösungen an.

Manager

Manager überwachen Mitarbeiter und den Ticketfortschritt.

Sie können Tickets neu zuweisen und Berichte erstellen.

Administrator (Optionale Rolle)

Verwalten die Systemeinstellungen, wie Benutzerrollen und Ticketkategorien.

Wichtige Objekte & Attribute:

1. **\*\*Ticket\*\***

ID (Eindeutige Kennung)

Titel (Kurze Beschreibung des Problems)

Beschreibung (Detaillierte Erklärung)

Status (Offen, In Bearbeitung, Gelöst, Geschlossen)

Priorität (Niedrig, Mittel, Hoch, Kritisch)

Erstellungsdatum

Letztes Aktualisierungsdatum

Zugewiesener Support-Mitarbeiter

## 2. **\*\*Benutzer (User) (Superklasse für Kunde, Support-Mitarbeiter und Manager)\*\***

BenutzerID

Name

E-Mail

Telefonnummer

## 3. **\*\*Kommentar (Comment)\*\***

KommentarID

Autor (Benutzer, der den Kommentar gepostet hat)

Nachricht

Zeitstempel

## 4. **\*\*Kategorie (Category)\*\***

KategorieID

Name (z. B. Technisches Problem, Rechnung, Allgemeine Anfrage)

## **\*\*Beziehungen:\*\***

Ein Kunde erstellt ein oder mehrere Tickets.

Jedes Ticket wird genau einem Support-Mitarbeiter zugewiesen.

Ein Manager kann mehrere Tickets überwachen.

Ein Ticket kann mehrere Kommentare haben.

Jedes Ticket gehört zu genau einer Kategorie.

Ein Support-Mitarbeiter kann mehrere Tickets bearbeiten, aber jedes Ticket hat nur einen zugewiesenen Mitarbeiter.

**\*\*Mögliche Methoden in den Klassen:\*\***

Ticket-Klasse Methoden:

WeiseMitarbeiterZu(Agent agent)

AktualisiereStatus(string neuerStatus)

FügeKommentarHinzu(Comment kommentar)

Benutzer-Klasse Methoden:

Einloggen()

Ausloggen()

ZeigeTickets()

Manager-Klasse Methoden:

WeiseTicketNeuZu(Ticket ticket, Agent neuerAgent)

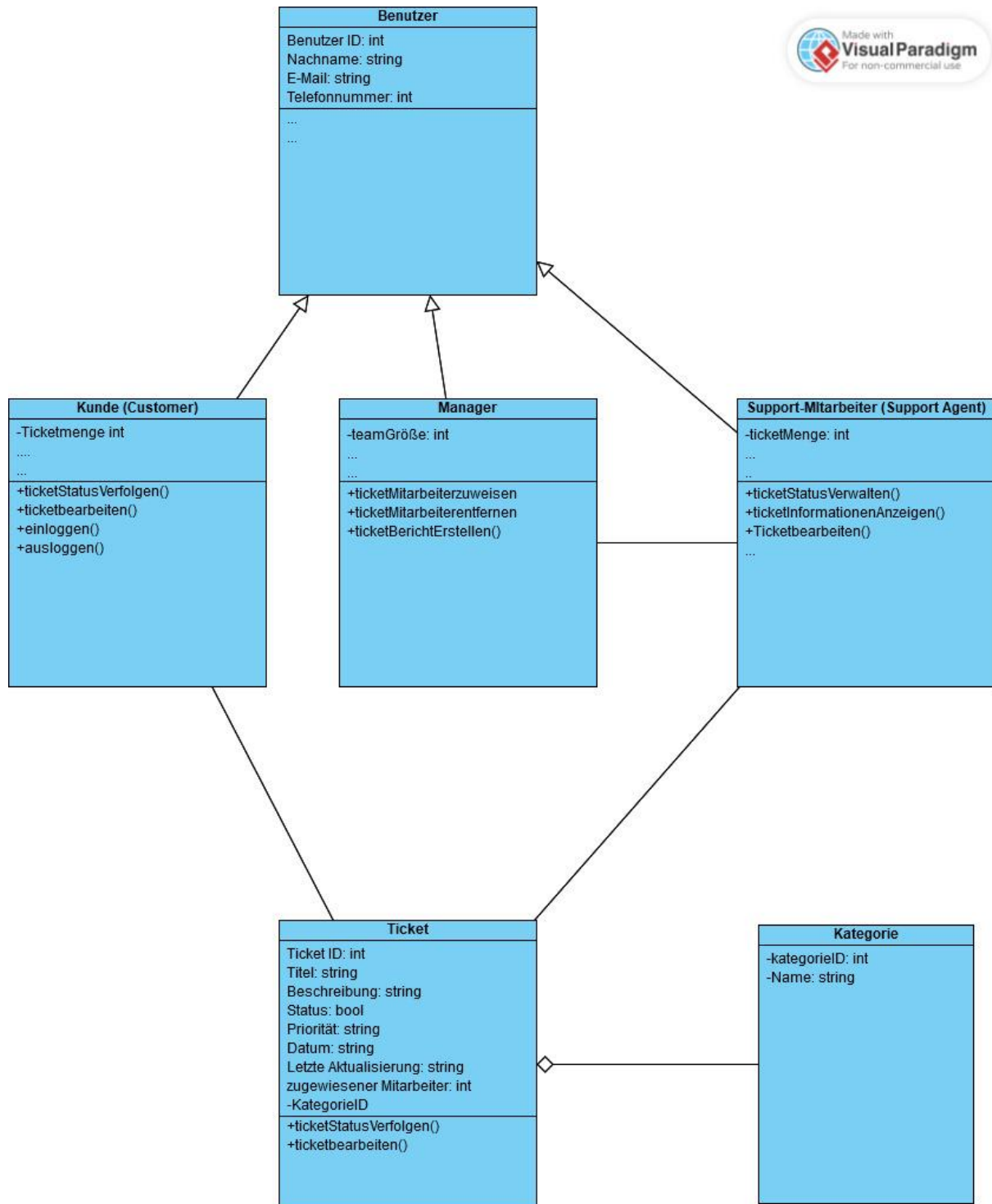
ErstelleBericht()

Hier das Bild von deinem Klassendiagramm hochladen:

---

**## Abgabe**

- Commit deine Dateien und pushen Sie diese auf euer Repository.
- Stelle auch sicher, dass all eure Referenzierte Bilder in euer Repository drin sind.



Viel Erfolg!