Klassen und Objekte

- *Einführung in Klassen und Objekte*

Zusammen:

```
class Person
{
    public string Vorname { get; set; }

    public string Nachname { get; set; }

    public int Alter { get; set; }


    public void SageEtwas(string satz)
    {
        Console.WriteLine(Vorname + " " + Nachname + ": " + satz);
    }
}
```

```
static void Main(string[] args)
{
    Person person1 = new Person();
    person1.Vorname = "Sabrina";
    person1.Nachname = "Müller";
    person1.SageEtwas("Hallo Leute!");
}
```

Aufgabe:

- Erstelle eine Klasse mit dem Namen Movie mit den Eigenschaften: Titel, Altersbeschränkung und Kategorie.
    - Erstelle eine zweite Klasse mit dem Namen MoviePlatform. Diese enthält die Main-Methode (Also den Einstiegspunkt deines Programmes).
      In der Main-Methode erstelle ein neues Objekt vom Typ Movie.
      Zum Schluss gib die Eigenschaften deines Films in der Konsole aus.

Objekteigenschaften

- *Einführung in Objekteigenschaften*

Zusammen:

```
private int anzahlTueren;
public int AnzahlTueren
{
    get
    {
        return anzahlTueren;
    }
    set
    {
        if (value > 0 && value <= 5)
            anzahlTueren = value;
    }
}
```

Aufgabe:

- Erstelle eine Klasse namens Computer. Diese besitzt die (privaten, also von außerhalb unzugänglichen) Variablen hdmiVorhanden und festplattenGröße. Des Weiteren verfügt die Klasse über die Funktionen SetzeHDMI(), SetzeFestplattenGröße() und ZeigeInfo().
    - Die Funktion SetzeFestplattenGröße akzeptiert nur Größen von 32GB bis 4TB
    - Erzeuge nun in deiner Klasse Program zwei Objekte vom Typen Computer. Weise diesen Werte zu und gebe die Infos dann in der Konsole aus.

Konstruktor und Dekonstruktor

- *Einführung in Konstruktor und Destruktor*

Zusammen:

OOP_Backup

Aufgaben:

Erstelle eine Klasse Student.

Diese Klasse hat folgende Attribute:

Name – Name des Studierenden
Studiengang– Den Studiengang der Person

Außerdem die Methode "Immatrikulieren"

Erstelle einen geeigneten Konstruktor.

Allgemeine Aufgaben zur OOP

Aufgaben:

- Erstelle eine Klasse für einen Fahrzeugverleih mit den Methoden zum Buchen, Abholen und Zurückgeben von Fahrzeugen. Diese Funktionen nehmen jeweils Objekte vom Typ "Fahrzeug" an.

Bankkonto

- Erstelle eine Klasse, die ein Bankkonto realisiert. Attribute für das Bankkonto sind der Name und Vorname des Kontoinhabers, die Kontonummer, der Kontostand sowie ein Limit, bis zu dem das Konto überzogen werden darf.
    - Erstellen einen oder mehrere geeignete Konstruktoren!
    - ErstellenMethoden für folgende Operationen:
        1. Ausgabe der Kontodaten (toString())
        2. Einzahlung.
        3. Auszahlung.
        4. Abfrage des Kontostandes.

Radio

https://tutego.de/javabuch/aufgaben/oop_classes.html

Radio

Implementiere eine Klasse Radio mit folgenden Attributen:

- eingeschaltet, wenn ein Radio an oder aus ist.
- lautstaerke, wie laut spielt das Radio Musik? (Die Lautstärke soll nur im Bereich von 0 bis 10 liegen.)
- frequenz, die die Frequenz des gewählten Senders angibt (Erlaubter Frequenzbereich ist zwischen 85.0 und 110.0).

Implementiere zu der Klasse zwei Konstruktoren der Art:

- Radio()
- Radio(boolean istAn, int lautstaerke, double frequenz)

Zu der Klasse Radio sollen folgende Methoden implementiert werden:

- lauter(), leiser(): Diese Methoden sollen die Lautstärke ändern (nur möglich im Zustand an).
- an(), aus(): Diese Methoden sollen den Zustand des Attributs eingeschaltet ändern.
- public String toString() Diese Methode soll Informationen über den internen Zustand als String zurückgeben. Es soll eine Zeichenkette der Form „Radio an: Freq=98.4, Laut=2“ zurückgeben.
- waehleSender(double frequenz) Diese Methode soll eine Frequenz speichern. Ist die gewählte Frequenz außerhalb des erlaubten Frequenzbereichs, so soll die Frequenz 99.9 gewählt werden.

Alle Methoden sollen sich selbst mitteilen. (Console.WriteLine())

Raumvermietung

Eine Firma vermietet für Gruppen, Vereine oder Privatpersonen eine Vielzahl von Räumen in einem Hochhaus. Der Firma gehört das gesamte Hochhaus, es stehen aber nicht immer alle Räume zur Verfügung. Um das Problem einfach zu halten, gehen wir davon aus, dass die zur Verfügung stehenden Räume durchnummeriert sind.

Für das Problem wurden folgende Parameter in einem Pflichtenheft festgehalten:

1. Von der Firma wird der Name, eine Adresse, die Anzahl der Räume zur Vermietung und die maximale Anzahl der möglichen Räume gespeichert.
2. Alle Attribute müssen gelesen werden können.
3. Die Größe der Räume (d. h. die Menge der Sitzplätze) wird in einem Feld gespeichert.
4. Die einzelnen Räume werden von Kunden gebucht, diese werden in einem Feld gespeichert.
5. Eine Firma wird mit ihrem Namen und der Adresse erzeugt, zudem werden alle notwendigen Werte gesetzt.
6. Es kann ein Raum (mit der Sitzplatzmenge) hinzugefügt werden.
7. Es muss eine Methode geben, die nach einem Raum mit einer gewünschten Größe sucht und dessen Nummer zurückgibt; findet sich kein Raum, wird -1 zurückgegeben.
8. Es muss eine Methode geben, die die Anzahl der freien (ungebuchten) Räume ermittelt und zurückgibt.

- Implementiere
    - den Konstruktor der Klasse Firma,
    - die Methode fuegeRaumHinzu(int plaetze),
    - die Methode sucheRaum(int plaetze) und
    - die Methode anzahlFrei().

Hotelreservierung

Aufgabe: Entwicklung eines Buchungs- und Reservierungssystems für ein Hotel in C#

Schritt 1: Erstelle eine Klasse "Room", die folgende Eigenschaften enthält:

- RoomNumber (int): die eindeutige Zimmernummer
- RoomType (string): der Typ des Zimmers (z.B. Einzelzimmer, Doppelzimmer, Suite)
- PricePerNight (double): der Preis pro Nacht für das Zimmer
- IsBooked (bool): gibt an, ob das Zimmer bereits gebucht ist oder nicht

Schritt 2: Erstelle eine Klasse "Guest", die folgende Eigenschaften enthält:

- Id (int): eine eindeutige Gäste-ID
- Name (string): der Name des Gastes
- CheckInDate (DateTime): das Datum des Check-Ins
- CheckOutDate (DateTime): das Datum des Check-Outs
- BookedRoom (Room): das gebuchte Zimmer

Schritt 3: Implementiere in der Hauptklasse das Buchungs- und Reservierungssystem, das es ermöglicht:

- Neue Gäste hinzuzufügen und Zimmer zu reservieren
- Überprüfen, ob ein bestimmtes Zimmer für einen bestimmten Zeitraum verfügbar ist
- Zimmer zu buchen und den entsprechenden Gästen zuzuweisen
- Preise für Reservierungen zu berechnen
- Informationen über gebuchte Zimmer und Gäste anzuzeigen

Hinweis: Verwende die DateTime-Klasse für die Datumsangaben und implementiere Methoden, um die Funktionalitäten des Systems zu realisieren.