

2. Wiederholung

2.1 Fragen

Gegeben ist folgender Beispielcode:

```
public class City : IKnowMyLocation, IKnowMyInhabitants
{
    public City(string name, Location location, int numberOfInhabitants)
    {
        Name = name;
        Location = location;
        NumberOfInhabitants = numberOfInhabitants;
    }

    public string Name { get; set; }
    public Location Location { get; set; }
    public int NumberOfInhabitants { get; set; }

    public static IEnumerable<City> CreateRandomly(int numberOfCities)
    {
        for (var i = 0; i < numberOfCities; i++)
        {
            var random = new Random();

            // create random identifier consisting of a letter and a digit
            var cityIdentifier = Convert.ToChar(random.Next(65, 91)).ToString()
                + random.Next(0, 10);

            yield return new City(
                $"City {cityIdentifier}",
                new Location
                {
                    Latitude = random.NextDouble() * 180 - 90,
                    Longitude = random.NextDouble() * 360 - 180
                },
                random.Next(10_000_000));
        }
    }

    public string TellLocation()
        => Location.TellLocation();

    public string TellNumberOfInhabitants()
        => $"I have {NumberOfInhabitants} inhabitants";
}

public class Location : IKnowMyLocation
{
    public double Latitude { get; set; }
    public double Longitude { get; set; }

    public string TellLocation()
    {
        var hemisphere = Latitude > 0 ? "northern" : "southern";
        var direction = Longitude > 0 ? "east" : "west";
        return $"I live in the {hemisphere} hemisphere, {direction} of
Greenwich.";
    }
}

public interface IKnowMyLocation
{
    public string TellLocation();
}
```

```
}  
  
public interface IKnowMyInhabitants  
{  
    public string TellNumberOfInhabitants();  
}
```

Kreuze an welche Aussagen zum Beispielcode korrekt sind:

- ☐ City und Location sind 2 Klassen
- ☐ Es können beliebig viele Objekte einer Klasse instanziiert (=erzeugt) werden, solange genügend Arbeitsspeicher dafür verfügbar ist.
- ☐ Zum Instanziiere eines Objekts einer Klasse wird das Keyword `out` verwendet.
- ☐ Zum Instanziiere eines Objekts einer Klasse wird das Keyword `new` verwendet.
- ☐ Zum Instanziiere eines Objekts der Klasse `City` müssen `name`, `location` und `numberOfInhabitants` in der Parameterliste des Konstruktors angegeben werden.
- ☐ Objekte der Klasse `City` können auch ohne Parameter im Konstruktor instanziiert werden, da immer ein Standardkonstruktor ohne Parameter automatisch erstellt wird.
- ☐ Klassen können von anderen Klassen abgeleitet werden. Eine abgeleitete Klasse erbt die Eigenschaften von ihrer Elternklasse.
- ☐ Zum Erstellen eines Objekts der Klasse `City` ist eine `Location` nötig. `City` hat somit die Eigenschaften von `Location` und ist von `Location` abgeleitet.
- ☐ Die Klasse `City` enthält eine statische Methode.
- ☐ Die Klasse `City` ist eine statische Klasse.
- ☐ Die Methode `CreateRandomly` der Klasse `City` erstellt mehrere Objekte (so viele wie durch den Parameter `numberOfCities` angegeben) des Typs `City` und gibt ihnen einen zufälligen Namen, zufällige Koordinaten und eine zufällige Einwohneranzahl.
- ☐ Da die Klasse `City` eine statische Klasse ist und von `Location` erbt, ist somit auch die Klasse `Location` eine statische Klasse.
- ☐ Zum Aufrufen der Methode `CreateRandomly` von außerhalb der Klasse `City`, muss zuvor ein Objekt der Klasse `City` instanziiert werden.
- ☐ Die Methode `CreateRandomly` kann von außerhalb der Klasse `City` aufgerufen werden, ohne vorher ein Objekt der Klasse `City` zu instanziiere.
- ☐ Eine statische Klasse wird durch das Keyword `static` bei der Klassendefinition gekennzeichnet und kann dadurch nur statische Data Members, statische Methoden und einen statischen Konstruktor enthalten.
- ☐ `City` und `Location` sind beide von `IKnowMyLocation` abgeleitet.
- ☐ `City` und `Location` implementieren beide das Interface `IKnowMyLocation`.
- ☐ Eine Klasse in C# kann von mehreren Elternklassen abgeleitet sein. Die Klasse `City` erbt von `IKnowMyLocation` und `IKnowMyInhabitants`. Das Konzept der Mehrfachvererbung ist nicht in allen Sprachen implementiert und in C# erst seit der Version 10, enthalten ab .NET 6, verfügbar.
- ☐ Lambda-Ausdrücke werden verwendet, um anonyme Funktionen zu erstellen. Die Funktionen sind anonym, weil sie im Gegensatz zu Methoden einer Klasse keinen Namen haben.
- ☐ Ein Lambdaausdruck kann verwendet werden, um Methoden-Bodies anzugeben, wie etwa bei `City.TellLocation`.