**Statische Klassen in C#**

**Multiple-Choice-Fragen (je 1 richtige Antwort)**

1. **Was ist eine statische Klasse in C#?**  
   a) Eine Klasse, die instanziiert werden kann  
   b) Eine Klasse, die nur einmal im Speicher existiert  
   c) Eine Klasse, die nur statische Mitglieder enthalten darf  
   d) Eine Klasse, die automatisch von Object erbt
2. **Was gilt für Konstruktoren in statischen Klassen?**  
   a) Sie sind nicht erlaubt.  
   b) Es ist nur ein privater Konstruktor möglich.  
   c) Sie können nur einmalig aufgerufen werden.  
   d) Sie müssen immer Parameter enthalten.
3. **Welche der folgenden Aussagen ist korrekt?**  
   a) Eine statische Klasse kann Interfaces implementieren.  
   b) Eine statische Klasse kann von einer Basisklasse erben.  
   c) Eine statische Klasse kann generisch sein.  
   d) Keine der oben genannten.
4. **Welche dieser Klassen ist in .NET eine statische Klasse?**  
   a) Math  
   b) Console  
   c) Array  
   d) File
5. **Wann verwendet man statische Klassen?**  
   a) Wenn mehrere Objekte unabhängig voneinander benötigt werden.  
   b) Wenn Methoden und Daten unabhängig von Objekten arbeiten sollen.  
   c) Wenn Polymorphie im Vordergrund steht.  
   d) Wenn dynamische Datenstrukturen gebraucht werden.
6. **Welcher Zugriff ist auf eine statische Klasse korrekt?**  
   a) new Math()  
   b) Math m = new Math();  
   c) Math.Sqrt(25)  
   d) var m = Math();

**Richtig/Falsch-Fragen**

1. Eine statische Klasse kann Objekte mit new erzeugen.  
   ✅ Richtig / ❌ Falsch
2. Eine statische Klasse kann Felder enthalten.  
   ✅ Richtig / ❌ Falsch
3. Eine statische Klasse kann sowohl statische als auch nicht-statische Methoden enthalten.  
   ✅ Richtig / ❌ Falsch
4. Statische Klassen eignen sich gut für Hilfsmethoden (Utility-Funktionen).  
   ✅ Richtig / ❌ Falsch
5. Eine statische Klasse wird automatisch als sealed betrachtet.  
   ✅ Richtig / ❌ Falsch
6. Statische Konstruktoren werden beim ersten Zugriff auf die Klasse ausgeführt.  
   ✅ Richtig / ❌ Falsch
7. Es kann nur einen statischen Konstruktor pro Klasse geben.  
   ✅ Richtig / ❌ Falsch
8. Eine normale Klasse mit ausschließlich statischen Methoden kann denselben Zweck erfüllen wie eine statische Klasse.  
   ✅ Richtig / ❌ Falsch

**Offene Fragen**

1. **Erklären Sie mit eigenen Worten, warum man eine statische Klasse nicht instanziieren kann.**
2. **Nennen Sie ein Beispiel aus der .NET-Bibliothek für eine statische Klasse und erklären Sie deren Nutzen.**
3. **Warum können statische Klassen nicht vererbt werden?**
4. **Erklären Sie den Ablauf eines statischen Konstruktors in einer statischen Klasse.**
5. **Nennen Sie ein Szenario, in dem Sie in einem eigenen Projekt eine statische Klasse sinnvoll einsetzen würden.**

**Lösungen: Quiz „Statische Klassen in C#“**

**Multiple-Choice-Fragen**

**1.** c) Eine Klasse, die nur statische Mitglieder enthalten darf  
**2.** c) Sie können nur einmalig aufgerufen werden (ein statischer Konstruktor wird beim ersten Zugriff ausgeführt)  
**3.** d) Keine der oben genannten  
**4.** a) Math (auch File, Path, Environment sind statisch, aber Console und Array nicht)  
**5.** b) Wenn Methoden und Daten unabhängig von Objekten arbeiten sollen  
**6.** c) Math.Sqrt(25)

**Richtig/Falsch-Fragen**

**7.** ❌ Falsch  
**8.** ✅ Richtig (aber nur statische Felder)  
**9.** ❌ Falsch (nur statische Methoden sind erlaubt)  
**10.** ✅ Richtig  
**11.** ✅ Richtig (statische Klassen sind automatisch sealed)  
**12.** ✅ Richtig  
**13.** ✅ Richtig  
**14.** ✅ Richtig (aber mit dem Unterschied, dass eine normale Klasse mit new instanziierbar wäre, auch wenn das keinen Sinn ergibt)

**Offene Fragen**

**15.**  
Weil statische Klassen nur statische Mitglieder enthalten und für den Zugriff kein Objekt notwendig ist. Ein Objekt hätte keine eigene Identität, daher ist eine Instanziierung verboten.

**16.**  
Beispiel: Math – liefert mathematische Funktionen wie Sqrt, Pow, Sin, die unabhängig von Objekten funktionieren.

**17.**

* **Statische Klasse**: Keine Instanz möglich, nur statische Mitglieder, rein globaler Zugriff.
* **Singleton-Klasse**: Normale Klasse, bei der genau eine Instanz erzeugt wird, auf die über eine statische Eigenschaft zugegriffen wird.

**18.**  
Weil statische Klassen automatisch sealed sind und keine Instanzen haben. Vererbung macht nur mit Instanzen Sinn.

**19.**  
Ein statischer Konstruktor wird **einmalig und automatisch** vor dem ersten Zugriff auf die Klasse ausgeführt (egal ob Feld, Methode oder Eigenschaft). Er dient typischerweise zur Initialisierung statischer Felder.

**20.**  
Beispiel-Szenario: Eine statische Klasse Währungsrechner mit Methoden wie EuroZuDollar(), DollarZuYen(), die unabhängig von Objekten arbeiten.