Wiederholung der Grundlagen

Die nachfolgenden kurzen Übungen dienen der Wiederholung der Grundlagen. Sie können alle Aufgaben in einem Projekt lösen, aber auch unterschiedliche Projekte verwenden. Sie sind sowohl eine Vorbereitung auf die Theorie- als auch auf die praktische Prüfung.

Operatoren

1. Sie starten mit der Variablen int i = 0; Erhöhen Sie die Variable um 1 und geben Sie das Ergebnis in der Konsole aus.
2. Erhöhen Sie die Variable nun um 3 und geben Sie das Ergebnis in der Konsole aus.
3. Multiplizieren Sie die Variable nun mit 3 und geben das Ergebnis in der Konsole aus.
4. Erhöhen Sie die Zahl um 20%.
5. Teilen Sie die Variable durch 4 und geben Sie sowohl das Ergebnis als Ganzzahl als auch den Rest in der Konsole aus.
6. Geben Sie in der Konsole aus: „Der aktuelle Wert der Zahl beträgt“ + Wert der Zahl.
7. Sie starten nun mit einer neuen Variablen string k = „Mein Text“; Geben Sie diesen in der Konsole aus.
8. Fügen Sie der Variablen „ ist kurz.“ hinzu. Geben Sie das Ergebnis in der Konsole aus.

Schleifen

1. For-Schleife
    a. Geben Sie 5 mal in der Konsole „Durchlauf:“ + die Nummer des Durchlaufs aus
    b. Wiederholen Sie diese Aufgabe, starten Sie diesmal aber mit der Zahl 5
    c. Wie oft wird diese Schleife ausgeführt? `for(int i=0, i<=5, i++) {}` Überlegen Sie erst und testen Sie es dann mithilfe eines kleinen Programms.

Dateizugriff

1. Schreiben Sie ein Programm, das über einen Button einen OpenFileDialog aufruft. Der Filter soll auf Textdateien gesetzt werden, das Startverzeichnis auf „Dokumente“.
2. Schreiben Sie ein Programm, dass eine Textdatei einliest und die Anzahl der Zeilen in der Konsole anzeigt. („Die Datei hat xy Zeilen“).
3. Lesen Sie alle Zeilen einer Datei ein und fügen Sie als Elemente einer Listbox hinzu.
4. Fügen Sie Ihrer Anwendung einen Button hinzu, der einen SaveFileDialog aufruft. Filter und Startverzeichnis sind identisch wie beim OpenFileDialog. Vergeben Sie als Standarddateinamen „MeineListe_Nummer.txt“. Die Nummer erhöhen Sie jedes Mal, wenn eine Datei abgespeichert wurde. Tipp: Speichern Sie diesen Zähler in einer globalen Variablen.
5. Speichern Sie Ihre Liste in dieser Datei.

Datum & Uhrzeit

1. Geben Sie das aktuelle Datum inkl. Uhrzeit in der Konsole aus
2. Geben Sie nur das aktuelle Datum aus.
3. Geben Sie nur die aktuelle Uhrzeit aus.

String-Formatierung

1. Legen Sie die Variable `double zahl = 12345.6789` an.
2. Geben Sie diese Zahl in der Konsole ohne Nachkommastelle aus.
3. Geben Sie diese Zahl mit genau zwei Nachkommastellen aus.
4. Geben Sie diese Zahl mit Tausender-Trennzeichen aus.
5. Geben Sie diese Zahl als Währung aus.
6. Geben Sie diese Zahl linksbündig mit einer Breite von 20 Zeichen aus.
7. Geben Sie diese Zahl rechtsbündig mit einer Breite von 20 Zeichen aus.

Fehlerbehandlung

1. Legen Sie in Ihrem Programm eine Textbox an und vergeben Sie einen passenden Namen.
2. Sobald der Benutzer in dieser Textbox die Enter-Taste drückt, prüfen Sie die Eingabe:
    a. Eingabe darf nicht leer sein
    b. Die Länge der Eingabe muss mindestens drei Zeichen sein
    c. Eingabe muss eine Zahl vom Typ int sein
    d. Die Zahl darf nicht kleiner 0 sein
    e. => Geben Sie im Fehlerfall eine Meldung über eine Messagebox aus. Andernfalls geben Sie die Zahl in der Konsole aus.

Methoden

1. Legen Sie in Ihrem Programm zwei Methoden „Addition“ und „Subtraktion“ an, die jeweils zwei Werte vom Typ int übernehmen und das Ergebnis als int zurückliefern.
2. Rufen Sie diese Methoden auf. Geben Sie das Ergebnis der zwei Operationen sowie den Rechenweg in der Konsole aus. Beispiel:
    a. Ausgabe für Addition: „Ergebnis der Addition: 2 + 3 = 5“

Objektorientierung

1. Erstellen Sie eine neue Klasse „Möbel“.
2. Diese Klasse besitzt eine Eigenschaft „Bezeichnung“, die nur im Konstruktor geschrieben werden darf, nach außen aber lesbar ist. Welches Schlüsselwort benötigen Sie?
3. Weiterhin besitzt es eine Eigenschaft „Preis“ vom Datentyp decimal. Diese Eigenschaft kann innerhalb der Klasse geschrieben und außerhalb nur gelesen werden. Wie legen Sie diese an? Wie heißt der zugehörige Codeausschnitt?
4. Erstellen Sie den Konstruktor, der als Argumente Bezeichnung und Preis übernimmt und diese Eigenschaften initialisiert.
5. Überschreiben Sie die ToString-Methode() sinnvoll.
6. Erstellen Sie in Ihrem Formular zwei Objekte vom Typ „Möbel“ und geben Sie diese in der Konsole aus.
7. Erstellen Sie eine Methode „PreisReduzieren“, die als Argument einen wert vom Typ int übernimmt und keinen Rückgabewert besitzt. Wie wird diese Methode angelegt? Welches Schlüsselwort sagt aus, dass eine Methode keinen Rückgabewert besitzt? Erstellen Sie die Methode und reduzieren Sie den Preis um den angegebenen Wert. Stellen Sie sicher, dass der Preis nicht negativ wird.

8. Fügen Sie die Eigenschaft „Farbe“ hinzu (Datentyp string). Diese soll von außen sowohl les- als auch schreibbar sein.
9. Erweitern Sie ihren Konstruktor, so dass auch diese Eigenschaft initialisiert wird.
10. Passen Sie Ihre ToString()-Methode so an, dass auch die Farbe ausgegeben wird.
11. Weisen Sie Ihren beiden Möbel-Objekten zwei unterschiedliche Farben zu und geben Sie beide Objekte in der Konsole aus.
12. Erstellen Sie drei weitere Möbel-Objekte.
13. Speichern Sie alle Möbel-Objekte in einer passenden Liste.
14. Geben Sie alle Objekte der Liste mithilfe einer geeigneten Schleife in der Konsole aus.
15. Setzen Sie bei zwei Objekten die Farbe auf „Gelb“.
16. Verwenden Sie LINQ, um eine neue (gefilterte) Liste zu erstellen, die nur die Objekte enthält, deren Farbe „Gelb“ ist.
17. Geben Sie diese Liste in der Konsole aus.
18. Erstellen Sie ein neues Formular „Möbelstück hinzufügen“. Darin soll ein neues Möbelstück angelegt werden. Es soll die Schaltflächen „Speichern“ und „Abbrechen“ besitzen.
19. Fügen Sie Ihrer Anwendung einen Button hinzu, mit dem Sie dieses Formular aufrufen und Ihrer Liste neue Möbelstücke hinzufügen können. Nach jedem Hinzufügen geben Sie folgendes in der Konsole aus:
    a. „Es wurde ein Objekt hinzugefügt:“ + Objektdetails
    b. „Objekte in der Liste:“ + Alle Objekte ausgeben (mit passender Schleife)
20. Implementieren Sie in Ihrer Klasse die Methoden Equals und GetHashCode (Hinweis: das geht automatisch!)
21. Implementieren Sie in Ihrer Klasse die Schnittstelle „IComparable<T>“. T steht dabei für Ihren Klassennamen – ersetzen Sie T also durch „Möbel“. IComparable soll für die Eigenschaft „Bezeichnung“ implementiert werden.
22. Geben Sie nun Ihre Liste sortiert nach der Bezeichnung in der Konsole aus.