

Häufig kommt es vor, dass in Formularen Eingaben nötig sind, die einem bestimmten Format entsprechen müssen. Ein einzugebendes Datum soll im Format [tt.mm.jjjj] vorliegen, für eine Kreditlaufzeit können nur Ganzzahlen akzeptiert werden oder für einen Geldbetrag sind nur Kommazahlen erlaubt. Hierfür stehen verschiedene Überprüfungsfunktionen bereit, die zusammen mit einer IF-Verzweigung angewendet werden.

1. Überprüfen, ob überhaupt eine Eingabe in das Textfeld gemacht wurde:

```
if (txtProdukt.Text == "")
{
    txtProdukt.Text = "Bitte eine Angabe machen...";
}
```

2. Überprüfen, ob die Eingabe die richtige Länge hat:

```
int laenge = txtProdukt.Text.Length;
if (laenge < 10)
{
    MessageBox.Show("Die Eingabe muss länger als 10 Zeichen sein.");
}
```

3. Überprüfen, ob die Eingabe ein bestimmtes Zeichen enthält:

```
string mail = txtMailKunde.Text;
bool enthalten = mail.Contains('@');
if (enthalten == false)
{
    MessageBox.Show("Die eMail enthält kein @.");
}
```

4. Überprüfen, ob es sich um den geforderten Datentyp Ganzzahl, Text oder Kommazahl handelt. Hier im Beispiel wird auf einen Ganzzahl-Typ überprüft.

```
int laufzeit = 0;
allesOK = int.TryParse(txtLaufZeit.Text, out laufzeit);
if (allesOK == false)
{
    MessageBox.Show("Es wurde keine ganze Zahl eingegeben.");
}
```

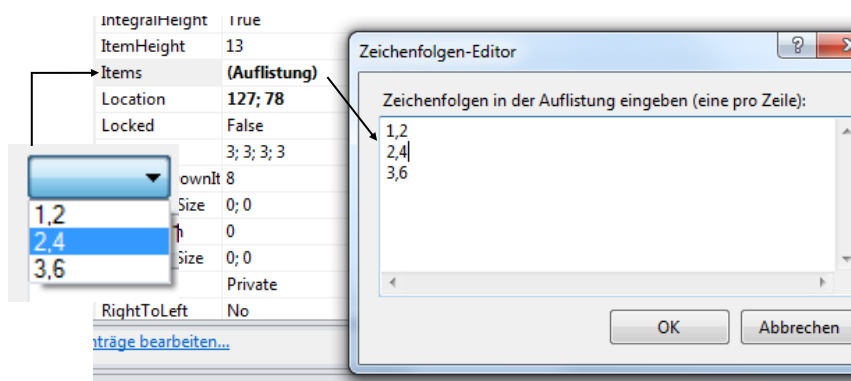
5. Überprüfen, ob die Eingabe einem speziellen *Muster* entspricht. Z. B. einer eMail mit einem '@' in der Mitte und einem Punkt vor dem Domänen-Angabe, wie zum Beispiel "berg@its-stuttgart.de"

```
string mail = txtMailKunde.Text;
if (Regex.IsMatch(mail, ".+@.+\.\.+") == false)
{
    MessageBox.Show("Die eMail wurde falsch eingegeben.");
}
```

Achtung:
Für Programmierprofis!!!

6. Verwendung von speziellen Oberflächen-Controls, wie *ComboBox* und *DateTimePicker*. Hier können die Werte nicht per Tastatur willkürlich eingegeben werden. Es stehen fest definierte Werte zur Auswahl. In einer ComboBox können diese auswählbaren Werte vorher durch den Programmierer festgelegt werden. Siehe hierzu die Eigenschaft *Items* im Formular-Entwurf. Auf den ausgewählten Text kann ebenfalls über die "Text"-Eigenschaft des Controls zugegriffen werden:

```
string datum = dtpAuslieferung.Text;
string taktFrequ = cboTakt.Text;
```



Ihre Aufgabe

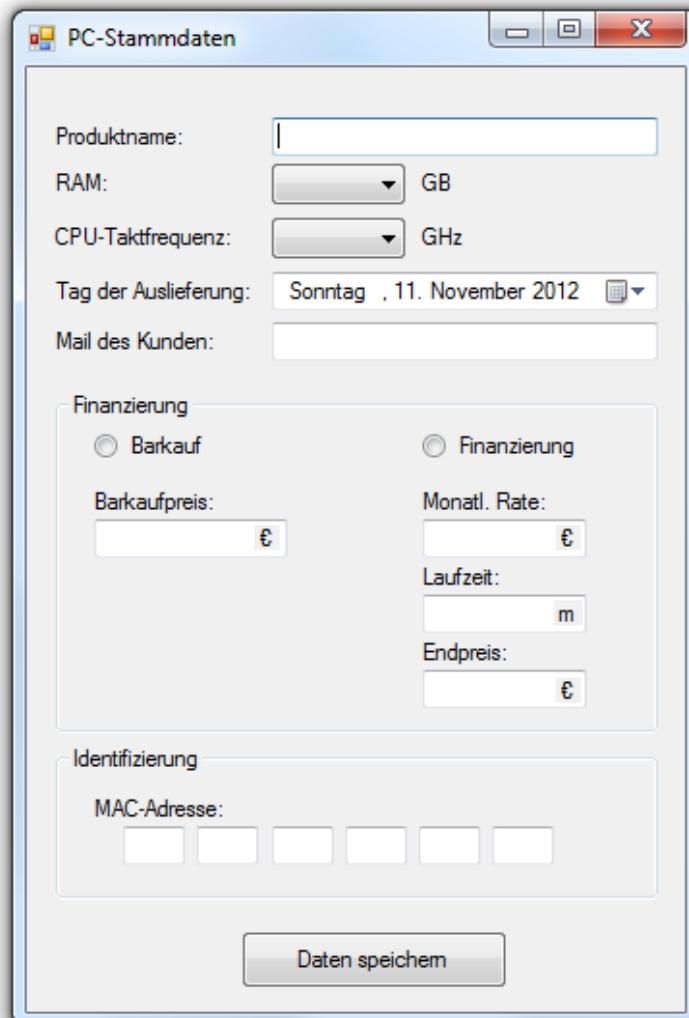
Untersuchen Sie Ihr eigenes Stammdaten-Formular. Für welche Eingabefelder würden Sie eine der oben genannten Überprüfungen einbauen? Begründen Sie Ihre Entscheidung.

Fügen Sie für folgende Eingabefelder eine Überprüfung ein:

Produktname, MAC-Adresse, Barkaufpreis, monatl. Rate, Laufzeit, Endpreis, Kunden-eMail,

Nutzen Sie den Event-Handler des "Daten-Speichern"-Buttons für die Überprüfung.

Geben Sie eine Fehlermeldung in Form einer MessageBox aus, wenn mindestens eine Eingabe nicht den Anforderungen entspricht.



The screenshot shows a Windows-style application window titled "PC-Stammdaten". The form contains the following fields and controls:

- Produktname:** A text input field.
- RAM:** A dropdown menu followed by the unit "GB".
- CPU-Taktfrequenz:** A dropdown menu followed by the unit "GHz".
- Tag der Auslieferung:** A date picker showing "Sonntag , 11. November 2012".
- Mail des Kunden:** A text input field.
- Finanzierung:** A section with two radio buttons: "Barkauf" and "Finanzierung".
 - Barkauf:** Includes a "Barkaufpreis:" label and a text input field with a "€" symbol.
 - Finanzierung:** Includes a "Monatl. Rate:" label and a text input field with a "€" symbol, a "Laufzeit:" label and a text input field with an "m" symbol, and an "Endpreis:" label and a text input field with a "€" symbol.
- Identifizierung:** A section with a "MAC-Adresse:" label and six small text input fields for hexadecimal digits.
- Daten speichern:** A button at the bottom of the form.