

# Lösungen

## 1. Das erste Projekt

### Fragen

1. Ein Programm in Visual Basic hört nicht einfach auf, wenn die letzte Programmzeile ausgeführt wurde. Das Formfenster bleibt weiter bestehen. Erst mit einem Klick auf Schließen oder den Tasten [Alt][F4] ist das Programm abgeschlossen.
2. In Visual Basic umfasst die Erstellung und Bearbeitung eines Programms gleich mehrere Dateien. Deshalb bezeichnet man alles zusammen als Projekt.
3. Im **Formular** verteilst und ordnest du deine Komponenten aus der Komponentenpalette (bis jetzt hast du nur die Schaltfläche benutzt). Im **Editor (Quelltextfenster)** kümmerst du dich um die Methoden, damit das Programm laufen kann. Ohne den Programm- bzw. Quelltext führt nämlich eine Aktion wie z.B. das Klicken auf eine Schaltfläche zu keinem erkennbaren Ergebnis.

### Keine Aufgaben

## 2. Buttons und Labels

### Fragen

1. Hauptkomponente ist das Formular, kurz `Form` genannt. Daneben sind aus diesem Kapitel erst mal zwei weitere Komponenten bekannt: Die Schalt- und die Anzeigefläche (Button und Label).
2. Ereignisse sind z.B. Mausklicks oder Tastendrucke. Mit Ereignissen lassen sich Methoden verknüpfen, die aktiviert werden, wenn ein Ereignis eintritt: So wird z.B. bei Mausklick die Methode `ButtonClick` aktiviert. Viele Methoden sind mit einem Objekt wie z.B. Formularen, Buttons, Labels oder anderen Komponenten verbunden.
3. Der Eigenschaft `Text` muss ein neuer Wert zugewiesen werden, z.B.

```
Button1.Text = "Lass mich!"  
Label1.Text = "Das ist mir egal!"
```

### Aufgabe

1. Einen Versuch findest du im Projektordner mit dem Namen HALLO1A.

## 3. Zahlen, Zeichen und Operatoren

### Keine Fragen

### Aufgaben

1. Das Projekt HOROSKOP1 bietet dir ein Gerüstprogramm mit zwölf beschrifteten Buttons, zu denen die `Button_Click`-Methoden bereits existieren. Nur die passenden Vorhersagetexte musst du selber finden.
2. Dazu schau mal in den Quelltext zum Projekt MATHE2.

## 4. Bedingungen

### Fragen

1. Für `Zensur` wird eine zufällige Ganzzahl zwischen 1 und 6 erzeugt.

### Aufgaben

1. Vorschläge dazu findest du in `MATHE1B` bzw. in `MATHE2A`.
2. Schau dir dazu mal das Projekt `MATHE3` an.

## 5. Ein Ratespiel mit Zahlen

### Frage

1. Der Haken ist, dass auch "zu groß" ausgegeben wird, wenn die Zahl richtig ist: Das Gegenteil (`Else`) von `< ist nämlich >=`.

### Aufgaben

1. Dieses *Raten*-Projekt trägt den Namen `RATEN3A`.
2. Dazu gibt es ein Angebot in `RATEN4`. (Wenn in anderen Bereichen als zwischen 0 und 10000 geraten werden soll, kannst du die Anfangswerte von *Oben* und *Unten* entsprechend ändern.)

## 6. Kontrollstrukturen

### Fragen

1. Das Programm steckt in einer Endlos-Schleife fest: Weil das Kapital sich nie vermehren kann, wird daraus nie eine Million. Heraus kommt man da aus Visual Basic nur über **DEBUGGEN** und **DEBUGGEN BEENDEN**.
2. Über die Eigenschaft `SelectedIndex` wird bei einem Listenfeld (`ListBox`) ermittelt, welcher Eintrag ausgewählt worden ist.

### Aufgaben

1. Eine Lösung steht in `MILLION2`.
2. Schau dir dazu das Projekt `HOROSKOP2` an. (Allerdings gibt's auch hier nur ein Gerüst.)
3. Probier mal das Projekt `DRUECK1` aus.

## 7. Combo, Radio oder Check?

### Frage

1. Wenn Optionsfelder oder Kontrollfelder in einem Formular zu einer Gruppe zusammengefasst sind, kann man bei Optionsfeldern (`RadioButtons`) immer nur eine Komponente einschalten. Bei Kontrollfeldern (`CheckBoxes`) dagegen lassen sich beliebig viele Komponenten einer Gruppe aktivieren.

## Aufgaben

1. Dazu findest du ein Gerüstprogramm im Projekt HOROSKOP3.

## 8. Aktion Seelenklempner

### Fragen

1. Anzeigefelder vom Typ `Label` können nur Text anzeigen, während man in Eingabefeldern vom Typ `TextBox` Text eingeben und bearbeiten kann.
2. Eine Komponente besitzt den Fokus, wenn sie als erste in einer »Ansammlung« von Komponenten aktivierbar bzw. »ansprechbar« ist. Will man beim Programmlauf einer Komponente den Fokus geben, muss man sie erst anklicken.
3. Man benötigt dazu sogenannte `Stream`-Objekte z.B. vom Typ `Reader` oder `Writer`. Das Laden und Speichern von Daten geschieht dann über `Read`- oder `Write`-Methoden (z.B. für Textzeilen über `ReadLine` und `WriteLine`).

### Aufgaben

1. Dazu musst du dir mal die Projektdatei `PROZENT1` anschauen.
2. Eine Lösung ist in der Methode `Button2_Click` des Projekts `KLEMP6A` zu finden.

## 9. Menüs und Dialoge

### Fragen

1. Zuerst legt man ein Symbol für die Komponente `MenuStrip` im Formularfenster ab. Mit Doppelklick auf den Eintrag "Hier eingeben" lassen sich dann Menütitel und Menüeinträge (Menüitems) eingeben. Nach Doppelklick auf einen Menüeintrag kann man die damit verknüpfte Methode bearbeiten.
2. Benötigt wird zunächst ein Objekt vom Typ `PrintDocument`, über damit verbundene `Print`-Methoden lässt sich u.a. Text (zeilenweise) über einen Drucker ausgeben. Die Schrift des Textes kann separat (`Font`) eingestellt werden. Für die eigentliche Textanzeige (bzw. das "Zeichnen von Text") ist ein Objekt vom Typ `Graphics` nötig, das dafür die Methode `DrawString` zur Verfügung stellt.

### Keine Aufgaben

## 10. Grafik

### Fragen

1. Der Ursprung (auch Null-Null-Punkt genannt) beginnt oben links in der Ecke der Anzeigefläche (z.B. Bildschirm)
2. Der erste Wert ist jeweils die Zeichenfarbe. Alle anderen Werte sind paarweise zu betrachten: Bei `DrawLine` geht es um den Start- und den Endpunkt einer Geraden. Bei `DrawRectangle` und `DrawEllipse` ist es eigentlich das Gleiche, nur "wandert" der Stift einen Umweg: statt "schräg" führt der Weg hier nur horizontal oder vertikal zum Ziel. Und `DrawEllipse` müht sich anschließend noch um eine möglichst große "Rundung" innerhalb eines (unsichtbaren) Rechtecks.

## Aufgaben

1. Die Lösung findest du im Projekt GRAFIK6A.
2. Eine (einfache) Lösung enthält das Projekt GRAFIK8.

## 11. Eine eigene Klasse

### Frage

1. Bei einer Klassenvereinbarung (mit `Class KlassenName`) wird sozusagen das Gerüst vereinbart: Eigenschaften und Methoden sind vorhanden. Aber erst nach einer Objektvereinbarung (mit `Dim Objekt As KlassenName` oder `Shared Objekt As KlassenName`) können die Eigenschaften mit Werten gefüllt und die Methoden zum Leben erweckt werden. Wichtig ist, dass das Objekts vor dem "Gebrauch" über den Konstruktor `New` initialisiert wird!

### Aufgabe

1. Dazu solltest du mal die Projektdatei MOVIE2B öffnen.

## 12. Kapselung und Vererbung

### Fragen

1. Daten und Code bzw. alle Eigenschaften und Methoden eines Objekts werden zu einer Einheit zusammengefasst. Ein Zugriff von außen ist nur über das Objekt direkt möglich – wenn überhaupt.
2. Wird eine Klasse so vereinbart `Class Name : Inherits Vorfahr`, dann verfügt sie automatisch über alle Eigenschaften und Methoden des Vorfahren. Mit einer geeigneten Basisklasse lässt sich so eine komplette Klassenfamilie aufbauen, bei dem die letzte Klasse in der Hierarchie alles erbt, was all ihre Vorfahren an Fähigkeiten zu bieten haben. (Voraussetzung für die Nutzung ist immer, dass es bei der Vererbung keine Einschränkung gibt – siehe dazu übernächstes Kapitel.)

### Aufgaben

1. Eine Möglichkeit findest du im Projekt MATHE4A.
2. Die Try-Catch-Strukturen stehen in MATHE4B.

## 13. Jetzt wird geOOPT

### Frage

1. Mit einer Zuweisung von `Nothing` wird ein Bild gelöscht, muss dann aber erneut geladen werden.

### Aufgaben

1. Getanzt und gehüpft wird im Projekt MOVIE7.
2. Eine Lösungsmöglichkeit findest du in MATHE6.

## **14. Polymorphie**

### Fragen

1. Zunächst scheinen beide gleichwertig zu sein. Aber nur `Overrides` macht es möglich, während der Laufzeit des Programms die passende Methode aufzurufen, wenn die Muttermethode mit `Overridable` virtuell vereinbart wurde.
2. Überschreiben kann man eine geerbte Methode durch eine neue. Auch überladene Methoden haben den gleichen Namen, sind aber in derselben Klasse definiert und unterscheiden sich z.B. in Typ oder Parameterliste. In der Regel ist hier der Zusatz `Overloads` angebracht.

### Keine Aufgaben

## **15. Buntes Allerlei**

### Frage

1. Bei Eingangsparametern wird deren Wert kopiert und innerhalb der Methode verarbeitet. Außerhalb ändert sich nichts, weil ja mit einer Kopie gearbeitet wurde. Bei Durchgangsparametern wird gar keine Kopie erzeugt, sondern über die Adresse direkt der Wert verarbeitet und damit auch außerhalb der Methode verändert.

### Aufgabe

1. Dazu schau dir mal das Projekt `PARAM1A` an.