

Übungsaufgaben (Blatt2)

- 8) Schreiben Sie ein Programm zur Umrechnung von Währungen als Windowsprogramm. Beachten Sie, dass bei fehlerhaften Eingaben (Buchstaben, Sonderzeichen) das Programm nicht abstürzen soll. Hier empfiehlt sich auf jeden Fall eine robuste Ausnahmebehandlung. Die Oberfläche sollte etwa nebenstehendes Aussehen besitzen. Die Eingabe in einem der 3 Eingabefelder führt zu einer sofortigen Anpassung eines anderen Feldes.



- 9) Schreiben Sie eine Funktion *Armstrong(zahl)*, die *true* zurückliefert, wenn die natürliche Zahl *zahl* eine Armstrongzahl ist, sonst *false*. Testen Sie diese Funktion durch Ausgabe aller 16 Armstrongzahlen zwischen 1 und 10000.  
Definition einer Armstrongzahl: Eine n-stellige Zahl heißt Armstrongzahl, wenn die Summe der n-ten Potenzen der einzelnen Ziffern der Zahl gleich der Zahl selbst ist.  
Beispiel: Die Zahl 153 hat drei Ziffern und es gilt  $1^3 + 5^3 + 3^3 = 153$ . Daher ist 153 eine Armstrongzahl.  
Literaturhinweis: Siehe unter Wikipedia (Suchbegriff: Armstrong Zahl, narcissistic number).  
Kurioses: Es gibt genau 88 Armstrongzahlen! Die größte Armstrongzahl hat 39 Stellen.
- 10) Schreiben Sie die Klasse *Matrix* als die Klasse der Matrizen. Wir benötigen die Operatoren +, -, \*, ebenso Funktionen zur Ein- und Ausgabe. Beliebige Matrizen sollten bearbeitet werden können. Ob intern mit ein- oder zweidimensionalen Feldern oder ausgefranzten Feldern gearbeitet wird, ist zweitrangig. Die Klasse ist soweit zu erstellen, dass in einem Programm Klassenobjekte mit beliebigen Matrizengrößen definiert und verwendet werden können. Natürlich ist zu beachten, dass nur passende Matrizen miteinander verknüpfbar sind. Schreiben Sie ein kleines, nicht in dieser Klasse befindliches Anwendungsprogramm zum Testen dieser Klasse.
- 11) Die Addition, Subtraktion und Multiplikation von Matrizen werden mittels eines Windowsprogramms visuell dargestellt. Die Klasse *Matrix* aus Aufgabe 9 ist zu verwenden. Das Formular enthält 2 quadratische 3x3-Matrizen, in die Werte eingegeben werden können. In einer dritten Matrix wird das Ergebnis ausgegeben. Mittels Radio-Buttons kann ausgewählt werden, welche Operation (Addition, Subtraktion, Multiplikation) auszuführen ist. Die beiden Eingabematrizen sind passend vorzubelegen.
- 12) Das Stackprogramm aus der Vorlesung vermischt einen Stack und die Oberfläche miteinander. Trennen Sie diese beiden Funktionalitäten. Erstellen Sie daher eine eigene Klasse *Stack*, die intern mit einer ArrayList arbeitet. In der Oberfläche wird dann ausschließlich auf Member und Eigenschaften dieser Klasse *Stack* zugegriffen. Ein direkter Zugriff auf die ArrayList ist nicht möglich. Fügen Sie zum Stackprogramm aus der Vorlesung zusätzlich ein *Popend* hinzu. Diese Methode liest das Element am Ende der Liste aus. Bauen Sie diese Funktion in die Klasse *Stack* und die Oberfläche mit ein.
- 13) Manchmal wird eine Erweiterung des Stackprogramms durch zusätzliche Statistik-Elemente gewünscht. Leiten Sie daher von der Klasse *Stack* aus Aufgabe 12 eine geeignete Klasse ab. Diese Klasse enthält die zusätzlichen Eigenschaften *Füllung* und *Anzahl*. Die Eigenschaft *Füllung* merkt die bisherige mittlere Füllung des Stacks, die Eigenschaft *Anzahl* zählt die Anzahl der Änderungen der Einträge im Stack seit Programmstart. Erweitern Sie das Programm weiter durch eine geeignete Anzeige dieser Eigenschaften auf der Oberfläche.
- 14) Das Schiebepuzzle aus der Vorlesung wird dahingehend modifiziert, dass statt Buttons jetzt 15 Bildteile verschoben werden, bis das korrekte Bild wiederhergestellt ist. Diese Aufgabe erfordert ein gewisses Umdenken, da jetzt tatsächlich die Teile zu verschieben sind und nicht nur der Button-Text geändert wird.
- 15) Schreiben Sie ein Programm, das auf die Bike-Datenbank in einer MySQL-Datenbank wie folgt zugreift. Es werden alle Mitarbeiter angezeigt (Relation Personal). Weiter kann in einer Combobox ein Mitarbeiter ausgewählt werden und in einem weiteren Feld die Gehaltserhöhung eingetragen werden. Durch Klicken eines Buttons wird dann diese Gehaltserhöhung für diesen Mitarbeiter ausgeführt. Sicherheitshalber sollte vor der Änderung des Gehalts die neue Gehaltssumme in einer MessageBox angezeigt werden. Erst bei der Bestätigung wird dieser Betrag in die Datenbank übernommen und natürlich auch sofort angezeigt.