

Aufgabe 2 – Phone Card

Teil 1: Klassische prozedurale Programmierung

1. Definieren Sie eine Variable ‚phoneCard‘ vom Typ int mit einem Startguthaben von 15,-- EUR, die überall bekannt ist.
Definieren Sie zudem eine Variable ‚phoneCardStatus‘ (ebenfalls vom Typ int). Diese Variable kann die Werte 0, 1 oder 2 annehmen, mit folgender Bedeutung:

0:	0,-- EUR oder 3,-- EUR Guthaben
1:	6,-- EUR oder 9,-- EUR Guthaben
2:	12,-- EUR oder mehr Guthaben

2. Schreiben Sie eine Prozedur ‚phoneCardMakeCall()‘, die das Guthaben der Telefonkarte pro Anruf/ Aufruf um 3,-- EUR verringert, solange der Wert dabei nicht kleiner als 0 wird – ein Telefonat ist nur möglich, wenn die Karte mindestens noch 3,-- EUR Guthaben aufweist. Das Guthaben darf nicht kleiner als 0 werden.
Weiterhin sorgt die Prozedur ‚phoneCardMakeCall()‘ dafür, dass die Statusvariable entsprechend oben genannter Regeln aktualisiert wird.
Benutzen Sie zur Implementierung keine if-Anweisungen!
3. Schreiben Sie eine Prozedur ‚phoneCardRefill_15()‘, die die Telefonkarte mit einem festen Betrag von 15,-- EUR auflädt und den Status auf 2 setzt.
4. Die Prozedur ‚phoneCardMakeCall()‘ gibt keine Informationen auf dem Bildschirm aus – das übernimmt ‚phoneCardDisplay()‘. Implementieren Sie diese Prozedur, die folgenden Text, abhängig von der Variable ‚phoneCardStatus‘ ausgeben soll:

0:	„Karte bitte auffuellen“,
1:	„Karte ist ca. halbvoll“,
2:	„Karte ist gut gefuellt“,

jeweils gefolgt vom tatsächlichen aktuellen Kartenguthaben zur Kontrolle.
Sie sollen ebenfalls keine if-Anweisungen verwenden.

5. Testen Sie Ihr Programm mit den folgenden Aufrufen in ‚main()‘:

```
phoneCardDisplay();  
  
phoneCardMakeCall();  
phoneCardDisplay();  
  
phoneCardMakeCall();  
phoneCardDisplay();  
  
phoneCardMakeCall();  
phoneCardDisplay();
```

```

phoneCardMakeCall();
phoneCardDisplay();

phoneCardMakeCall();
phoneCardDisplay();

phoneCardMakeCall();
phoneCardDisplay();

phoneCardRefill_15();
phoneCardDisplay();

phoneCardRefill_15();
phoneCardDisplay();

```

6. Was ist die Ausgabe Ihres Programmes? Geben Sie hierzu wieder einen entsprechenden Screendump ab.

Teil 2: Objektorientierte Programmierung:

1. Deklarieren Sie eine Klasse CPhoneCard in der Datei ,PhoneCard.h'. Sie habe neben dem Konstruktor (parameterlos; initial 15,-- EUR Startguthaben) drei Methoden, analog zu oben genannten Prozeduren: ,makeCall()', ,refill_15()' und ,display()'. Die Attribute sind ,mCredit' und ,mStatus'.
2. Implementieren Sie die Klasse in der Datei ,PhoneCard.cpp'. Die Methoden sollen dieselbe Logik haben wie in Teil 1.
3. in einer ,PhoneCardOOMain.cpp' – Datei definieren Sie drei Objekte der Klasse CPhoneCard: Eines im Datensegment, eines auf dem Stack und eins auf dem Heap. Für die Objekte im DS und auf dem Stack zeigen Sie jeweils einen exemplarischen Methodenaufruf, z.B. ,heapObject.display()'. Für das Objekt auf dem Heap rufen Sie zum Testen die Methoden analog zum Test im 1. Teil der Aufgabe auf.
4. Wie groß ist ein Objekt der Klasse CPhoneCard?
5. Was ist die Ausgabe Ihres Programms (Abgabe Screendump).

Viel Spaß und viel Erfolg.