Vererbung: Zahlenlose

In diesem Beispiel geht es um *Glücksspiel-Lose*.

Hinweise:

- Verwende in allen Konstruktoren *Initialisierungslisten*!
- Du darfst den gesamten Methoden-Code *innerhalb* des **`class`** schreiben.
- Alle Konstruktoren und Methoden sind *öffentlich*.

a) Klasse `Los`

Diese Klasse enthält nur, was den Losen aller derartigen Glücksspiele *gemeinsam* ist:

- Ein *Member* für die *Losnummer* (**`int`**)
  und ein Member für den *Kundennamen* (**`string`**),
  beide auch für abgeleitete Klassen, aber nicht für fremden Code sichtbar.
- Einen *Konstruktor*. Er wird mit dem Kundennamen als Parameter aufgerufen
  und speichert ihn im Member.
  Weiters gibt er dem Los *automatisch* die fortlaufend nächste Nummer.

  Das *Member*, das du zur *Erzeugung der fortlaufenden Nummern* brauchst
  (wie muss es dafür deklariert werden?), soll *nur innerhalb* der Klasse sichtbar sein.
- Die *virtuelle Methode* **`gewinnRang`** zur Berechnung des Gewinns:
  Sie hat *keinen Parameter* und liefert einen **`int`**-Returnwert:
  0 … kein Gewinn, 1 … kleinster Gewinn, 2 … zweitkleinster Gewinn usw.

  Diese Methode muss in jeder abgeleiteten Klasse anders *überschrieben* werden
  und bleibt daher in der Klasse **`Los`** ausdrücklich *unimplementiert*
  (daher dürfen Objekte der Klasse **`Los`** auch *nicht* direkt erzeugbar sein).
- Was musst du noch (nicht rein virtuell, sondern virtuell und mit leerem Code)
  deklarieren, wenn eine Klasse virtuelle Methoden hat?

b) Klasse `ZahlenLos`

Diese Klasse ist von **`Los`** *abgeleitet* und implementiert Lose,
bei denen man auf *eine einzige Zahl* wettet (wie bei Glücksspirale oder Spiel 77).

- Definiere dazu zuerst einmal zwei *programmweite* **`int`**-*Konstanten*:
  - Die *Anzahl der Ziffern* **`anzZiffern`**, die diese Zahl hat (nimm 7).
  - Die *größtmögliche Zahl* **`maxZahl`** (bei 7 Ziffern 9999999).
  Achtung: Unter *Windows* solltest du höchstens 4 und 9999 nehmen, da das **`rand`**
  unter Windows nur Zahlen bis **`32767`** liefert (und nicht bis 2 Mrd. wie bei Linux).
- Jedes Objekt enthält (sichtbar für abgeleitete Klassen) *zwei* **`int`**-*Member*:
  - Eines für die *Zahl*, auf die mit diesem Los *gewettet* wurde.
  - Ein *klassenweites Member* für die *Zahl*, die in der *aktuellen Ziehung gezogen* wurde.
- Der *Konstruktor* hat *zwei Parameter*:
  - Der *Kundenname* (**`string`**) wird an **`Los`** *weitergegeben*.
  - Die *gewettete Zahl* (**`int`**) wird im Member *gespeichert*.
- Schreib eine *klassenweite Methode* **`Ziehung`** *ohne* Parameter und Returnwert,
  die eine Ziehung macht: Sie *speichert* im Klassen-Member für die gezogene Zahl
  eine *Zufallszahl* zwischen 0 und **`maxZahl`** (*einschließlich!*).

- **gewinnRang** wird wie folgt *überschrieben*:

  Zähle in einer *Schleife* mit **anzZiffern** Durchläufen, wie viele *Ziffern* der gewetteten Zahl *gleich* der entsprechenden Ziffer der gezogenen Zahl sind (bei uns zählen *alle* gleichen Ziffern, nicht nur die am hinteren Ende der Zahl), und gib diese Anzahl richtiger Ziffern als *Returnwert* zurück.

  Tipp:

  - Kopiere beide Zahlen vor der Schleife in *zwei Hilfsvariablen*.
  - Was musst du mit dem Ergebnis-*Zähler* vor der Schleife tun?
  - *Vergleiche* in der Schleife die *letzte Ziffer* beider Hilfsvariablen (mit welcher Rechenoperation erhältst du die letzte Ziffer einer Zahl?) und zähle eins zum Ergebnis dazu, wenn sie gleich sind. *Streiche* dann von beiden Hilfsvariablen die *letzte Ziffer* weg (wie?).

c) Klasse ZahlenLosStat

Diese Klasse ist von **ZahlenLos** abgeleitet. Sie *funktioniert genauso* wie **ZahlenLos**, aber speichert im Objekt zusätzlich *Statistiken über alle Gewinne* dieses Loses.

- Die Klasse hat *zwei* nur innerhalb der Klasse sichtbare *Member*:
  - Einen **int**-Wert für die *Anzahl* der **gewinnRang**-*Aufrufe* dieses Objektes.
  - Ein **int**-Array mit **anzZiffern+1** vielen Elementen, das im **i**-ten Element zählt, *wie oft* der **i**-te Gewinnrang gewonnen wurde: Der kleinste Rang ist 0, der höchstmögliche Gewinnrang ist **anzZiffern**: Alle Ziffern richtig.
- Es gibt einen *Konstruktor* mit *denselben Parametern* wie in **ZahlenLos**. Er reicht die Parameter an den **ZahlenLos**-*Konstruktor* weiter und *setzt* dann den Aufrufszähler und alle Array-Elemente *auf 0*.
- **gewinnRang** wird nochmals *überschrieben*: Es ermittelt den Gewinnrang durch Aufruf von **gewinnRang** der *Vaterklasse*, erhöht den *Aufrufszähler* und das dem Gewinnrang entsprechende *Array-Element* um 1, und gibt den Gewinnrang dann als *Returnwert* zurück.
- Beim *Abbau* (Löschen) eines **ZahlenLosStat**-Objektes sollen *automatisch* dessen *Statistiken* auf **cout** *ausgegeben* werden: Eine Zeile mit Losnummer, Kundenname, gewetteter Zahl und Anzahl der Ziehungen, und eine Zeile, wie oft das Los wie viele Ziffern richtig hatte (Werte im Array mit einer Schleife ausgeben).

d) Hauptprogramm

Schreib zum Testen folgendes **Hauptprogramm** (*ohne* Angaben auf der Befehlszeile):

- Sorge dafür, dass bei jedem Programmlauf *andere Zufallszahlen* berechnet werden.
- Leg zwei **Los**-*Pointer* an und lass sie auf zwei *dynamisch angelegte* **ZahlenLosStat**-Objekte zeigen: "Franz" wettet auf 1234567 und "Xaver" setzt 6666666.
- Mach eine *Schleife* mit 1 Million Durchläufen. Ruf in jedem Durchlauf zuerst die Klassen-Methode **Ziehung** auf und dann je ein Mal **gewinnRang** für jedes unserer beiden Lose.
- Gib beide Objekte wieder *frei*, damit deren *Statistiken* ausgegeben werden.