Hochschule Worms
University of Applied Sciences
Fachbereich Informatik
Prof. Dr. Bernd Ruhland

Objektorientierte Programmierung

Übungsaufgabe Typtransformation

Casting mal wörtlich: „**PSDS P**raktikum **s**ucht **d**en **SuperStar**“

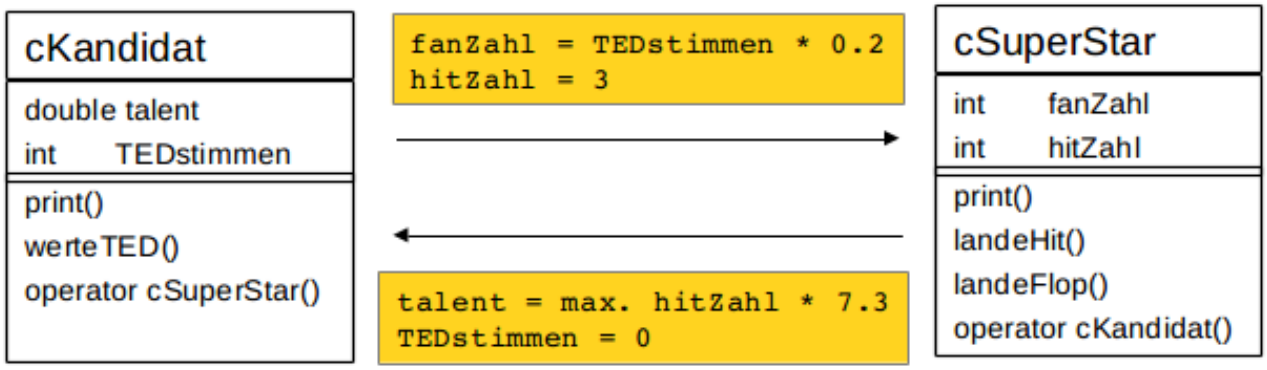
Typumwandlung für Kandidaten und SuperStars:

Lernziele:

Typumwandlung zwischen Klassen
Cast-Operator
Anforderungen erkennen und einhalten

Hinweis:

Der Aufgabentext ist im generischen Femininum gehalten. Natürlich sind andere Geschlechter wie z.B. Kandidaten und männliche Superstars genauso gemeint



Motivation:

Eine Kandidatin hat Talent. In die Casting Show kommt sie mit einem unterschiedlichen Startvolumen an Talent, im Normalfall sind es **10.0** Talenteinheiten.

Aus einer Kandidatin kann in einer Casting-Show ein SuperStar werden, wenn sie bei einem TED (Telefonabstimmung) **mehr als 100 000 Punkte** bekommt. Die Punkte ergeben sich aus der Zahl der TED-stimmen multipliziert mit dem Talent.

Wird sie zum SuperStar dann werden aus den TED-stimmen der Kandidatin die Anzahl ihrer Fans berechnet, aber nur mit dem **Faktor 0.2** (man nimmt an, dass alle Fans im Schnitt 5 mal beim TED angerufen haben).

Ein SuperStar landet Hits oder Flops. Aus der Casting-Show bringt sie bereits 3 Hits mit wenn sie zum SuperStar wird. Ein Flop mit **landeFlop()** reduziert die Zahl der Hits um 1, ein Hit mit **landeHIT()** erhöht die Zahl um 1. Wenn die Anzahl der Hits unter 2 fällt, wird der SuperStar wieder eine Kandidatin.

In diesem Fall bestimmt die Maximalzahl der Hits (neues Attribut!), die der SuperStar jemals hatte, das neue Talent: Das Talent ist die maximal erreichte Hitzahl multipliziert mit dem **Faktor 5.3**.

Aufgabe:

Erstellen Sie die Klasse „**cKandidat**“:

- Privates Attribut **talent** vom Typ double, Vorgabewert 10.0.
- Privates Attribut **TEDstimmen** vom Typ int, Vorgabewert 0.
- Universeller Konstruktor aus Double-Wert und Integer-Wert.
- Ausgabemethode „***print()***“ für Talent und TED-Stimmen.
- **Cast-Operator cSuperStar** zur Typkonvertierung **cKandidat --> cSuperStar**. Die Werte der Attribute des Zielobjekts errechnen sich aus den Werten der Attribute des Ausgangsobjekts, siehe Motivation. Beachten Sie, dass Punkte nicht gleich Stimmen sind.
- Eine Methode „***int werteTED (int stimmen)*“**, die den Wert des Attributs TEDstimmen setzt und die Anzahl der Wertungspunkte zurückgibt: Die Wertungspunkte berechnen sich aus dem Aufrufparameter Anzahl der Stimmen multipliziert mit dem Talent.

Erstellen Sie die Klasse „**cSuperStar**“:

- Privates Attribut **fanZahl** vom Typ int, Vorgabewert 0.
- Privates Attribut **hitZahl** vom Typ int, Vorgabewert 0.
- Universeller Konstruktor aus zwei int-Werten.
- Ausgabemethode „**print()**“ für die Anzahlen der Fans und Hits.
- **Cast-Operator cKandidat** zur Typkonvertierung **cSuperStar --> cKandidat**. Die Werte der Statusvariablen des Zielobjekts errechnen sich aus den Werten der Statusvariablen des Ausgangsobjekts, siehe Motivation.
- Eine Methode ***landeHit()***, die die Hitzahl um 1 erhöht.
- Eine Methode ***landeFlop()***, die die Hitzahl um 1 erniedrigt.

Respektieren Sie die Kapsel. Erstellen Sie bei Bedarf Hilfsmethoden. Ergänzen Sie bei Bedarf Attribute.

Erstellen Sie ein Hauptprogramm:

- Definieren Sie eine Instanz „susanne“ der Klasse cKandidat.
- Definieren Sie eine Instanz „suzie“ der Klasse cSuperStar.
- Geben Sie die Werte der beiden Objekte mittels der print()-Methoden aus.
- Führen Sie solange TED-Aufrufe durch, bis der Kandidat die Punktegrenze erreicht hat, geben Sie jeweils die Werte des Kandidaten aus.
- Weisen Sie dann dem cSuperStar-Objekt das „gecastete“ cKandidat-Objekt zu, verwenden Sie dazu die Typkonvertierung, die Sie erstellt haben:

```
suzy = (cSuperStar)susanne;
```

- Geben Sie danach die Werte des cSuperStar-Objektes aus.
- Landen Sie Hits und Flops wie es Ihnen gefällt.
- Fällt der Hit-Wert unter 2, dann weisen Sie dem Objekt cKandidat das „gecastete“ Objekt cSuperStar zu, verwenden Sie dazu die Typkonvertierung, die Sie erstellt haben:

```
susanne = (cKandidat)suzy;
```

- Geben Sie danach die Werte des cKandidat-Objektes noch mal aus.