

C++ Übungsblatt 1: Klassen

Aufgabe 1-1: Erste Klasse

Erstellen Sie eine Klasse *Rectangle*, mit deren Hilfe Rechtecke in der Konsole angezeigt werden (Sie erinnern sich vielleicht an Aufgabe L6 zurück). Gehen Sie hierzu wie folgt vor:

1. Erstellen Sie das Grundgerüst der Klasse
2. Fügen Sie der Klasse die Attribute *width* und *height* hinzu (wählen Sie hierzu einen angemessenen Datentyp)
3. Schreiben Sie einen Konstruktor, der mit den zwei Parametern *_width* und *_height* seine Attribute initialisiert. Die Unterstriche sind dort, um Verwechslungen zwischen den Attributen und den Parametern zu vermeiden
4. Erstellen Sie eine Methode *void draw()* in der Klasse, die das entsprechende Rechteck auf die Konsole ausdrückt

Folgender Programmcode sollte ausführbar sein, sofern sich die Klasse in einer Headerdatei namens *rectangle.h* befindet.

```
1 #include "rectangle.h"
2
3 int main()
4 {
5     Rectangle rect(4, 5);
6     rect.draw();
7
8     rect.width = 2;
9     rect.height = 10;
10    rect.draw();
11
12    return 0;
13 }
```

Sie können Ihre Klasse nun durch Methoden mit weiteren Funktionalitäten ausstatten.

- Die Methoden *grow()* und *shrink()* sollen das Rechteck jeweils um 1 vergrößern bzw. verkleinern. Achten Sie hierbei, dass sich die Größe bei der unteren bzw. oberen Schranke nicht verkleinert bzw. vergrößert
- Die Methode *resize(double factor)* ändert die Größe des Rechtecks um einen entsprechenden Faktor
- Die Methode *area()* gibt die Fläche des Rechtecks zurück, *circumference()* den Umfang
- Es soll angegeben werden können, ob vom Rechteck nur der Rand oder auch eine ausgefüllte Fläche ausgedruckt werden soll. Sie können diese Einstellungen durch beliebige Muster erweitern
- ...

Aufgabe 1-2

Eine Reiseagentur will verschiedene Angebote für Urlaubsreisen von Kunden bewerten lassen. Die Urlaubsziele wurden im System bisher mit Ort, Dauer und dem Preis eingespeichert.

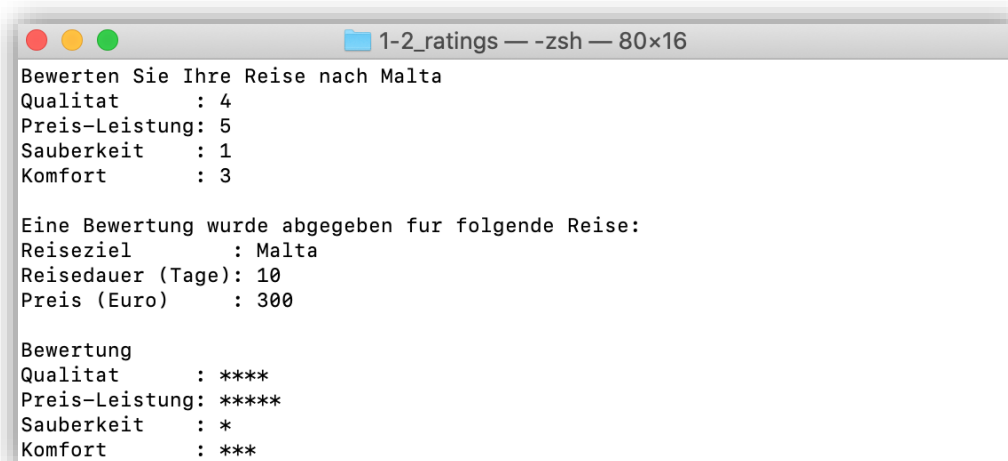
```
1 #include <string>
2
3 class Trip
4 {
5 public:
6     // name of the location
7     std::string destination;
8
9     // duration in days
10    unsigned int duration;
11
12    // price in euros (no cents)
13    unsigned int price;
14
15    Trip(std::string _dest, unsigned int _duration, unsigned int _price)
16        : destination(_dest), duration(_duration), price(_price)
17    {
18        // everything already initialized, nothing to do here
19    }
20 };
```

Um nun die Bewertung hinzuzufügen, soll eine neue Klasse *Rating* erstellt werden. Diese beinhaltet die folgenden Kategorien:

- Qualität
- Preis-Leistungsverhältnis
- Sauberkeit
- Komfort

Jede Kategorie kann mit einem Wert von 1 (schlecht) bis 5 (exzellent) bewertet werden.

Anschließend kann der Klasse *Trip* ein Attribut der Klasse *Rating* hinzugefügt werden. Testen Sie Ihr Ergebnis, indem Sie den Benutzer nach einer Bewertung fragen und diese danach wieder ausgeben. Das kann in etwa wie folgt aussehen



```
1-2_ratings --zsh-- 80x16
Bewerten Sie Ihre Reise nach Malta
Qualität      : 4
Preis-Leistung: 5
Sauberkeit    : 1
Komfort       : 3

Eine Bewertung wurde abgegeben fur folgende Reise:
Reiseziel     : Malta
Reisedauer (Tage): 10
Preis (Euro)  : 300

Bewertung
Qualität      : ****
Preis-Leistung: *****
Sauberkeit    : *
Komfort       : ***
```

BONUS

Um das Feature komplett zu machen, muss es nun nur noch möglich sein, für eine Reise mehrere Bewertungen abzugeben. Erstellen Sie hierzu eine Methode in *Trip*, mit der Ratings hinzugefügt werden können. Eine weitere Methode sollte dazu dienen, um das Durchschnitts-Rating zurückzugeben.