

Praktikumstutorium – Blatt 3

Als Vorbereitung auf das Testat 3 sollten Sie unbedingt diese Aufgaben bearbeiten.

Sofern dabei Schwierigkeiten auftreten, sollten Sie unbedingt das **Tutorium des Programmierpraktikums** besuchen

Tutoriumszeiten im Raum OH 12/4.030

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
12.00 – 14.00 Uhr	10.00 – 16.00 Uhr	10.00 – 14.00 Uhr	10.00 – 14.00 Uhr	Testat

1 - Klasse Box

Schreiben Sie eine Klasse **Box**, die eine Kiste in Form eines Quaders beschreibt. Ein Quader ist ein geometrischer Körper mit sechs rechteckigen Flächen, bei dem alle Winkel 90 Grad haben. Ein Quader lässt sich durch drei Angaben spezifizieren: Breite, Höhe und Tiefe. Nutzen Sie dazu reelle Zahlen des Typs **double**.

Schreiben Sie für die Klasse **Box** verschiedene Methoden:

- ~~einen Konstruktor~~, der die Breite, Höhe und Tiefe setzt,
- ~~drei Methoden~~, die die Breite, Höhe und Tiefe zurückgeben (`getWidth`, `getHeight`, `getDepth`),
- ~~drei Methoden~~, die das Volumen (`getVolumeSize`), die Oberfläche (`getAreaSize`) und die Summe der Kantenlängen (`getEdgesLength`) berechnen und zurückgeben,
- ~~Die Methode `boolean isCube()`~~ soll den Wert **true** zurückgeben, falls es sich um einen *Würfel* handelt, also alle Kanten die gleiche Länge besitzen.
- ~~Die Methode `int compareTo(Box f)`~~ dient dazu, zwei Kisten miteinander zu vergleichen, das aufrufende Objekt und das als Parameter übergebene Objekt. Als Ergebnis soll zurückgegeben werden:
 - einen Wert größer als 0, falls die aufrufende Box ein größeres Volumen als die als Parameter übergebene Box besitzt,
 - der Wert 0, falls beide Kisten das gleiche Volumen besitzen,
 - einen Wert kleiner als 0, falls die aufrufende Box ein kleineres Volumen als die als Parameter übergebene Box besitzt.
- eine Methode ~~`encloses`~~, die bestimmt, ob eine als Parameter übergebene Kiste bei **parallel liegenden** Seitenflächen vollständig in das aufrufende Objekt **echt** eingepasst werden kann, und einen entsprechenden Wahrheitswert zurückliefert. Beachten Sie bei der Implementierung, dass Kisten gedreht werden können: Eine Kiste *30x20x10* passt echt in eine Kiste *11x31x21*.

2 - Klassen Period, PointInTime und Date

Klasse Period

Definieren Sie eine Klasse **Period**, die als einziges Attribut eine Zeitangabe in Minuten verwaltet. Eine negative Dauer soll nicht möglich sein. Das Attribut `minutes` soll nicht-öffentlich angelegt werden.

Programmieren Sie:

- Zwei ~~Konstrukturen~~, die mit einem Parameter (nur Minuten) bzw. zwei Parametern (Minuten und Stunden) ausgestattet sind. Bei negativen Parameterwerten soll 0 zur Initialisierung verwendet werden.
- Die Methode ~~`getMinutes`~~, die als Rückgabewert die Anzahl der abgelegten Minuten liefert.
- Die Zugriffsmethode ~~`getHours`~~, die die Zahl der in den abgelegten Minuten enthaltenen ganzen Stunden zurückgibt.
- Die Zugriffsmethode ~~`getMinorMinutes`~~, die den Wert der nicht in ganzen Stunden enthaltenen Minuten angibt, so dass gilt: `getHours()*60+getMinorMinutes()==minutes`.
- Eine Methode ~~`toString`~~, die die Dauer als Text der Form *hh:mm* zurückgibt.
- Eine Methode ~~`clone`~~, die eine *Kopie* des **Period**-Objektes erstellt und zurückgibt. Beachten Sie, dass hierzu das Erstellen eines neuen Objekts mit gleichen Inhalten notwendig wird.
- Eine Methode ~~`change`~~, die eine als Parameter übergebene Anzahl von Minuten zu der Dauer hinzuaddiert. Wird dieser Methode ein negativer Wert übergeben, so soll nichts geschehen.

Testen Sie Ihre Klasse **Period**, indem Sie mit den Konstruktoren **Period**-Objekte anlegen und für diese die Methoden aufrufen.

Klasse PointInTime

Ein Zeitpunkt soll aus drei nicht-öffentlichen Attributen bestehen:

- einer Jahresangabe,
- der Nummer eines Tages im Jahr und
- einer Stundenangabe.

Die Jahresangabe soll immer positiv und vierstellig (also größer als 999 und kleiner als 10000) sein. Schaltjahre werden nicht berücksichtigt, so dass der Tag zwischen 1 und 365 und die Stundenangabe zwischen 0 und 23 liegen dürfen. Die Klasse PointInTime soll einen geeigneten **Konstruktor** und **Methoden** zum Zurückgeben der drei Attribute bereitstellen. Zudem soll die Klasse folgende Methoden anbieten:

- Eine Methode ~~toString~~, die den Zeitpunkt als Text der Form *jjjj/ttt/hh* zurückgibt.
- Eine Methode ~~clone~~, die eine *Kopie* des PointInTime-Objektes erstellt und zurückgibt.
- Eine Methode ~~change~~, die eine als Parameter übergebene Anzahl von Stunden zu dem Zeitpunkt hinzuaddiert. Wird dieser Methode ein negativer Wert übergeben, so soll nichts geschehen.

Testen Sie Ihre Klasse PointInTime, indem Sie mit den Konstruktoren Period-Objekte anlegen und für diese die Methoden aufrufen.

Klasse Date

Ein Termin besteht aus einer Bezeichnung, einem Startzeitpunkt und einer Dauer. Implementieren Sie die Klasse Date unter Zuhilfenahme der Klassen PointInTime und Period. Alle Attribute sollen nicht-öffentlich angelegt werden.

Die Klasse Date soll einen **Konstruktor** mit je einem Parameter für die Bezeichnung, den Zeitpunkt und die Dauer, eine **Methode** zum Kopieren eines Termins und **Methoden** zum Zurückgeben der drei Attribute bereitstellen.

Zusätzlich soll die Klasse Date folgende Methoden bieten:

- Eine Methode ~~change~~, die einen Termin ändert. Diese Methode soll zwei Parameter des Typs **int** besitzen, von denen der erste die Anzahl der Stunden angibt, um die der Startzeitpunkt des Termins verschoben werden soll, und der zweite die Anzahl der Minuten angibt, um die sich die Dauer ändern soll.
- Eine Methode ~~toString~~, die eine geeignete Darstellung der Attribute des Termins als **String** zurückgibt.

Klasse TestAllClasses

- Legen Sie eine Klasse `TestAllClasses` und in dieser eine statische Methode `main` an.
- Erzeugen Sie in der Methode `main` ein Objekt der Klasse `Period` und von diesem eine Kopie.
- Ändern Sie die Dauer dieser Kopie und lassen Sie sich den Inhalt beider Objekte anzeigen.
- Erzeugen Sie ein Objekt der Klasse `PointInTime`, legen Sie wieder eine Kopie an und ändern Sie diese.
- Erzeugen Sie anschließend mit den zuvor erzeugten Objekten zwei unterschiedliche Objekte der Klasse `Date` und ändern Sie einen Termin. Überprüfen Sie die Attributwerte.