CLAUDE-DORNIER-SCHULE	Hausaufgabenplaner Kapselung Getter und Setter	Datum:

Wiederholung zur Datenkapselung

Schaue dir das Video zur Datenkapselung an: https://www.youtube.com/watch?v=haaJN5gewZA

Getter und Setter verwenden

Die Idee der Kapselung besteht darin, die Implementierung der Attribute vor dem Nutzer des Objektes zu verbergen. Sie sind dann privat. Ein Aufruf führt zwangsläufig zu einem Fehler, da das Attribut privat ist und aus dem Hauptprogramm nicht aufgerufen werden kann. [1)]

1. Aufgabe

Setze die Sichtbarkeit aller Attribute der Klasse Aufgabe auf `private`. (Hinweis: Dadurch entstehen Fehlermeldungen in der Klasse PlanerGUI – diese zunächst einfach ignorieren.)

Um den Zugriff auf die Attribute dennoch sicherzustellen, erfolgt dieser über eine get- und set-Methode, die zunächst definiert werden müssen.
Getter und Setter sind in der objektorientierten Programmierung spezielle Methoden, die ein Attribut abfragen oder ändern. Sie sind Teil der öffentlichen Schnittstelle eines Objekts und verbergen Implementierungsdetails dieses Objekts. Dadurch kann die Implementierung eines Objekts geändert werden, ohne seine öffentliche Schnittstelle zu ändern.

Der Getter

- Der Getter fragt ein Attribut ab und liefert dessen Wert
- Der Getter kommt beim Abrufen eines Werts zum Einsatz, zum Beispiel bei
  `tfTitel.setText(dieAufgabe.titel);`
- Aufbau am Beispiel einer Variable x mit beliebigen Datentyp:

```
public datentyp getX () {
  return x;
}
```

Der Setter

- Der Setter ändert oder setzt den Wert eines Attributs, also als Alternative zu z.B. `dieAufgabe.titel = "Programmieren üben";`
- Aufbau am Beispiel für eine Variable namens x mit beliebigen Datentyp:

```
public void setX (datentyp pX) {
 x = pX;
}
```

2. Aufgabe

Erstelle für alle Attribute zunächst einfache Getter und Setter nach dem oben gezeigten Schema in der Klasse Aufgabe. (Hinweis: Die Fehlermeldungen in der Klasse PlanerGUI bleiben weiterhin – diese zunächst einfach ignorieren.)

3. Aufgabe

Korrigiere die Fehler in der Klasse PlanerGUI.java, indem du alle Zugriffe auf die Attribute durch die Aufrufe des Getters bzw. Setters ersetzt. Beachte dabei folgendes Schema:

- **Der Getter – Lesender Zugriff**
  Statt `dieAufgabe.titel` schreibst du nun `dieAufgabe.getTitel(`. (Gleich für alle Attribute.)

- **Der Setter – Schreibender Zugriff**
  Dieser hat die Form, dass links vom Gleichheitszeichen ein Ausdruck in Punktnotation für den Zugriff auf ein Attribut steht, z.B. `dieAufgabe.titel = "Programmieren üben";`

  Mit einem Setter gibt es keine Zuweisung, also kein = mehr. Stattdessen wird der Setter - eine Methode ohne Rückgabewert - aufgerufen. Der Wert, der bisher rechts vom Gleichheitszeichen steht, wird als Parameter übergeben, also
  `dieAufgabe.setTitel("Programmieren üben");`

1) https://www.schule-bw.de/faecher-und-schularten/mathematisch-naturwissenschaftliche-faecher/informatik/material/materialien-zum-neuen-bildungsplan-wirtschaftsinformatik

Setter mit Funktionen

In der Grundform sind die Getter und Setter eigentlich unnötig. Um dies zu umgehen, werden die Setter nun mit zusätzlichen Funktionen versehen.
Theoretisch sind auch Getter mit zusätzlicher Funktion möglich, das ist jedoch meist bei abgeleiteten Attributen der Fall (siehe nächster Aufgabenteil).

4. Aufgabe
Verändere den Setter für den Titel: Dieser darf nicht mehr als 20 Zeichen haben. Ein überlanger Titel wird einfach auf 20 Zeichen gekürzt.

5. Aufgabe
Verändere die Setter für Tag und Monat in der Klasse Aufgabe: Der Tag muss ein Wert zwischen 1 und 31 sein, der Monat ein Wert zwischen 1 und 12. Werden ungültige Daten angegeben, wird der Wert stattdessen auf 1 gesetzt (oder: bei zu kleinem Wert auf 1, bei zu großem Wert auf 31 bzw. 12).

Wenn du möchtest: Kombiniere `setTag` und `setMonat` in einer Methode `public void setDatum (in pTag, int pMonat)` - zusätzlich zu den beiden einzelnen Settern oder anstelle derselben. In dieser Methode kannst du nun auch prüfen, ob der gewünschte Tag im gewünschten Monat existiert und so z.B. Daten wie 30.2. oder 31.9. vermeiden.

Der Konstruktor

6. Aufgabe
Erstelle einen Konstruktor für die Klasse Aufgabe. Dieser soll wie folgt aufgebaut sein:

```
public Aufgabe (String pTitel, String pFach, int pTag, int pMonat, String pBeschreibung)
```

Hinweis: Der Wert „erledigt“ wird nicht als Parameter übergeben.

7. Aufgabe
Ersetze anschließend in der Klasse PlanerGUI die Objekterstellung mit Standardkonstruktor und anschließendem Setzen der Werte durch eine Variante, die diesen Konstruktor nutzt.

Tipps / Hinweise:

- Die Setter für `fach` und `beschreibung` enthalten keine weiteren Prüfungen. Hier kannst du die übergebenen Werte direkt dem Attribut zuweisen.
- Rufe bei `titel` sowie `tag` und `monat` die Setter auf, um den Code für die Prüfung / Anpassung nicht doppelt zu haben. Das geht innerhalb der gleichen Klasse "ganz normal", also z.B. mit `setTitel(pTitel);` oder indem in der Punktnotation `this` vorangestellt wird, also `this.setTitel (pTitel);`
- `erledigt` wird immer auf `false` gesetzt.
- Wenn dir die Zeile mit dem Aufruf des Konstruktors zu lang wird, darfst du gerne mit "Zwischenvariablen" zu arbeiten.

Abgeleitetes Attribut: Überfällige Aufgaben

Kapselung beschränkt sich nicht auf Getter und Setter. Mit ihr sind beispielsweise auch abgeleitete Attribute möglich. Hier soll ein solches Attribut `uberfallig` eingeführt werden. Konkret: Bei Aufruf der Methode `getUberfallig` wird das aktuelle Datum abgefragt und mit dem Fälligkeitsdatum der Aufgabe verglichen. Die Aufgabe ist überfällig, wenn

- der Fälligkeitsmonat kleiner ist als der aktuelle Monat
- die Monate gleich sind, aber der Fälligkeitstag kleiner als der aktuelle Tag

8. Aufgabe

Implementiere die oben beschriebene Methode `public boolean getUberfallig ()` in der Klasse Aufgabe. Rufe die Methode bei der Anzeige einer Aufgabe auf. Ist die Aufgabe überfällig und nicht erledigt, soll das Textfeld des Titels rot eingefärbt werden. Andernfalls bleibt das Textfeld weiß.

Benötigte Befehle

Den aktuellen Tag und aktuellen Monat bekommst du mit folgenden Befehlen:

```
int aktTag = java.time.LocalDate.now().getDayOfMonth();
int aktMonat = java.time.LocalDate.now().getMonthValue();
```

Die Farbe eines Textfelds kannst du mit `setColor` setzen:

```
tfTitel.setColor(Color.RED);
```

bzw. `tfTitel.setColor(Color.WHITE);`