

Zeiterfassungsbericht erstellen

Ausgangssituation

Das Unternehmen Code-Like-A-Pro möchte zukünftig die Arbeitszeiten seiner Mitarbeiter aufzeichnen. Dazu soll ein Zeiterfassungssystem eingeführt werden.

Wenn ein Mitarbeiter zur Arbeit kommt, meldet er sich am System an. Das System vermerkt dann die aktuelle Uhrzeit in einer speziellen Erfassungsdatei, deren Format in den folgenden Abschnitten noch erläutert wird. Hat der Mitarbeiter Feierabend, meldet er sich wieder vom System ab. Auch hier speichert das System die aktuelle Uhrzeit in der Erfassungsdatei.

Für jeden Mitarbeiter und jeden Monat werden neue Erfassungsdateien erstellt und in einem zentralen Verzeichnis abgelegt. Der Name einer Erfassungsdatei setzt sich aus drei Feldern zusammen: Jahr, Monat und Personalnummer des Mitarbeiters. Die Felder werden durch Bindestriche getrennt. Die Dateiendung ist `.txt`. Beispiel: `2024-03-EF0123.txt`. Eine Personalnummer besteht ausschließlich aus Buchstaben und Ziffern. Ihre genaue Zusammensetzung ist für diese Aufgabe irrelevant.

Eine Erfassungsdatei enthält keine, eine oder beliebig viele Zeilen / Einträge. Für eine Zeile gilt:

- Eine Zeile darf leer sein bzw. nur aus Leerzeichen bestehen.
- Nicht leere Zeilen setzen sich zusammen aus den drei Feldern `Tag`, `Stunden` und `Minuten`, jeweils getrennt durch Semikolons.
- Alle anderen Zeilen deuten auf fehlerhafte Daten hin.

Hier ein Beispiel für die Datei `2024-03-ef001.txt`:

```
1; 08; 15

1; 12; 05
2; 09; 00
3; 09; 10
5; 08; 50
5; 16; 45
7; 21; 15

7; 06; 15
28; 16; 45
31; 07; 45
```

Folgende Informationen lassen sich unter anderem aus dieser Datei entnehmen:

- Am 01.03.2024 kam der Mitarbeiter um 08:15 Uhr auf Arbeit.
- Am 01.03.2024 machte der Mitarbeiter um 12:05 Uhr Feierabend.
- Am 02.03.2024 begann der Mitarbeiter seine Arbeit um 09:00.
- Da ein zweiter Eintrag für den 02.03.2024 fehlt, wissen wir nicht, wann er Feierabend machte.

- Für den 04.03.2024 gibt es keine Einträge. Hier war der Mitarbeiter also abwesend.

Es werden folgende Annahmen gemacht:

- Für jeden Tag des Monats gibt es keinen, einen oder zwei Einträge.
- Wenn für einen Tag zwei Einträge existieren, so gibt der erste den Arbeitsbeginn an und der zweite den Feierabend.
- Falls nur ein Eintrag existiert, interpretieren wir ihn immer als Arbeitsbeginn.
- Wenn keine Einträge für einen Tag existieren, gehen wir davon aus, dass der Mitarbeiter an diesem Tag nicht gearbeitet hat.
- Wenn die Uhrzeit des zweiten Eintrages vor der Uhrzeit des ersten Eintrages liegt, arbeitet der Mitarbeiter über Nacht. In seiner Arbeitszeit ist also ein Tageswechsel enthalten. Die Arbeitsdauer wird jedoch vollständig auf den Vortag angerechnet.

Ein Beispiel für die Berechnung der Arbeitsdauer "über Nacht": Wenn die Schicht des Mitarbeiters am Tag 3 um 21:15 Uhr beginnt und erst am Folgetag um 06:00 Uhr endet, zeichnet das System folgende Einträge auf:

```
03; 21; 15
03; 06; 00
```

Es erfolgt in diesem Fall kein Eintrag für Tag 4. Die Arbeitsdauer für Tag 3 beträgt 8h 45m.

Aufgabenstellung

Das Code-Like-A-Pro Unternehmen hat dich damit beauftragt, eine Konsolenanwendung zu entwickeln, die eine Erfassungsdatei einliest und deren Daten formatiert ausgibt. Zusätzlich sollst du folgende Statistiken berechnen:

- Die Arbeitsdauer in Stunden und Minuten pro Tag.
- Die Gesamtarbeitsdauer des Monats in Stunden und Minuten.
- Die Anzahl der Tage, an denen der Mitarbeiter vollständig präsent und vollständig abwesend war.
- Die Anzahl der Tage, für die nur ein Eintrag existiert, also eine Feierabend-Zeit fehlt.
- Die Anzahl an Einträgen in der Erfassungsdatei.

Dein Programm soll abwärtskompatibel zu alten JDK Versionen sein. Aus diesem Grund musst du dich **auf folgende Datentypen beschränken**:

- Primitive Datentypen und deren Wrapper-Klassen
- Arrays
- Die Klasse Arrays
- String, StringBuilder
- Scanner, Path
- IOException
- StandardCharsets

Du darfst also insbesondere **keinen Gebrauch von Collections und Streams machen**. Du kannst deine Ausgaben mit `System.out` und `System.err` erzeugen.

Für dein Programm gelten weitere Einschränkungen: Dein Programm soll auf einem System laufen, das über sehr begrenzte Hardware-Ressourcen verfügt. **Aus diesem Grund ist es nicht möglich, die gesamte Erfassungsdatei in den Arbeitsspeicher zu laden bzw. deren Einträge in einen Array zu speichern.** Du musst also die Erfassungsdatei zeilenweise mit einem Scanner abarbeiten und hast damit **keinen wahlfreien Zugriff auf die Einträge**. Bildlich gesprochen: Du erhältst die Einträge auf einem Transportband - Eintrag für Eintrag, aber hast keine Möglichkeit, alle bisher entnommenen Einträge zwischenspeichern. Du "siehst" immer nur deinen gerade entnommenen Eintrag und den Eintrag, der als nächstes folgt.

Du hast großes Glück: Dein Kollege hat bereits mit der Entwicklung begonnen und die oben genannten Klassen importiert. Außerdem hat er schon alle notwendigen Methoden vordefiniert und mit Kommentaren versehen. Deine Aufgabe ist es nun, diese Methoden zu implementieren. **Beachte jedoch die Coding-Guidelines deines Teams:**

- Der vorgegebene Quelltext muss exakt übernommen werden. Umbenennungen sind untersagt.
- Zusätzliche Imports sind unzulässig.
- Es dürfen zusätzliche Konstanten hinzugefügt werden.
- Zusätzliche Methoden sind zulässig.
- Es ist SNAKE_CASE für Konstanten zu verwenden und camelCase für Methoden und Variablen.
- Hungarian Notation wie `iCount` oder `sName` ist untersagt.
- Es müssen aussagekräftige Bezeichner verwendet werden. Auf Abkürzungen wie `idx` oder `rng` ist zu verzichten. Ausnahme sind Laufvariablen wie `i` und `j` bei Schleifen.

Du kannst dir zu Testzwecken weitere Erfassungsdateien anlegen. **Das oben beschriebene Format dieser Erfassungsdateien darf jedoch nicht verändert werden.**

Das Programm soll wie folgt aufgerufen werden: `TimeSheetReport <filename>`, wobei `<filename>` tatsächlich nur ein Name wie `2024-01-ef005` ist, also kein Pfad und keine Dateiendung! Das Programm bildet aus dem Namen und der Dateiendung `txt` und dem Datenverzeichnis `time-sheets` einen Pfad.

Wie dein Programm die Daten formatieren soll, kannst du im Abschnitt [Programmausgaben](#) nachlesen.

Weitere Hinweise kannst du den Kommentaren im Quelltext entnehmen. Viel Erfolg!

Programmausgaben für die mitgelieferten Beispieldateien

Die folgenden Ausgaben helfen dir dabei, die Korrektheit deines Programms zu prüfen.

Für die Zeiterfassungsdatei `2024-03-ef001` erzeugt das Programm folgende Ausgabe:

```
=====
YEAR: 2024 / MONTH: 03 / ID: EF001
=====

DATE      | START | END   | DURATION | REMARKS
-----
01.03.2024 | 08:15 | 12:05 | 03h 50m  | fully present
02.03.2024 | 09:00 | n/a   | n/a      | clarification needed
03.03.2024 | 09:10 | n/a   | n/a      | clarification needed
04.03.2024 | n/a   | n/a   | n/a      | absent
05.03.2024 | 08:50 | 16:45 | 07h 55m  | fully present
```

06.03.2024	n/a	n/a	n/a	absent
07.03.2024	21:15	06:15	09h 00m	fully present
08.03.2024	n/a	n/a	n/a	absent
09.03.2024	n/a	n/a	n/a	absent
10.03.2024	n/a	n/a	n/a	absent
11.03.2024	n/a	n/a	n/a	absent
12.03.2024	n/a	n/a	n/a	absent
13.03.2024	n/a	n/a	n/a	absent
14.03.2024	n/a	n/a	n/a	absent
15.03.2024	n/a	n/a	n/a	absent
16.03.2024	n/a	n/a	n/a	absent
17.03.2024	n/a	n/a	n/a	absent
18.03.2024	n/a	n/a	n/a	absent
19.03.2024	n/a	n/a	n/a	absent
20.03.2024	n/a	n/a	n/a	absent
21.03.2024	n/a	n/a	n/a	absent
22.03.2024	n/a	n/a	n/a	absent
23.03.2024	n/a	n/a	n/a	absent
24.03.2024	n/a	n/a	n/a	absent
25.03.2024	n/a	n/a	n/a	absent
26.03.2024	n/a	n/a	n/a	absent
27.03.2024	n/a	n/a	n/a	absent
28.03.2024	16:45	n/a	n/a	clarification needed
29.03.2024	n/a	n/a	n/a	absent
30.03.2024	n/a	n/a	n/a	absent
31.03.2024	07:45	n/a	n/a	clarification needed

WORKING TIME : 20h 45m				
PRESENT : 03				
ABSENT : 24				
INCOMPLETE : 04				
RECORDS : 10				

Für die Zeiterfassungsdatei 2024-02-ef001 lautet die Ausgabe:

=====				
YEAR: 2024 / MONTH: 02 / ID: EF001				
=====				
DATE	START	END	DURATION	REMARKS

01.02.2024	n/a	n/a	n/a	absent
02.02.2024	n/a	n/a	n/a	absent
03.02.2024	n/a	n/a	n/a	absent
04.02.2024	n/a	n/a	n/a	absent
05.02.2024	n/a	n/a	n/a	absent
06.02.2024	n/a	n/a	n/a	absent
07.02.2024	n/a	n/a	n/a	absent
08.02.2024	n/a	n/a	n/a	absent
09.02.2024	n/a	n/a	n/a	absent

10.02.2024		n/a		n/a		n/a		absent
11.02.2024		n/a		n/a		n/a		absent
12.02.2024		n/a		n/a		n/a		absent
13.02.2024		n/a		n/a		n/a		absent
14.02.2024		n/a		n/a		n/a		absent
15.02.2024		n/a		n/a		n/a		absent
16.02.2024		n/a		n/a		n/a		absent
17.02.2024		n/a		n/a		n/a		absent
18.02.2024		n/a		n/a		n/a		absent
19.02.2024		n/a		n/a		n/a		absent
20.02.2024		n/a		n/a		n/a		absent
21.02.2024		n/a		n/a		n/a		absent
22.02.2024		n/a		n/a		n/a		absent
23.02.2024		n/a		n/a		n/a		absent
24.02.2024		n/a		n/a		n/a		absent
25.02.2024		n/a		n/a		n/a		absent
26.02.2024		n/a		n/a		n/a		absent
27.02.2024		n/a		n/a		n/a		absent
28.02.2024		n/a		n/a		n/a		absent
29.02.2024		n/a		n/a		n/a		absent

WORKING TIME : 00h 00m								
PRESENT : 00								
ABSENT : 29								
INCOMPLETE : 00								
RECORDS : 00								

Für die Datei 2024-01-ef005 ist die Ausgabe wie folgt:

=====

YEAR: 2024 / MONTH: 01 / ID: EF005

=====

DATE		START		END		DURATION		REMARKS

01.01.2024		n/a		n/a		n/a		absent
02.01.2024		n/a		n/a		n/a		absent
03.01.2024		13:25		n/a		n/a		clarification needed
04.01.2024		n/a		n/a		n/a		absent
05.01.2024		12:05		18:23		06h 18m		fully present
06.01.2024		10:45		n/a		n/a		clarification needed
07.01.2024		n/a		n/a		n/a		absent
08.01.2024		10:20		n/a		n/a		clarification needed
09.01.2024		07:50		n/a		n/a		clarification needed
10.01.2024		06:00		14:30		08h 30m		fully present
11.01.2024		n/a		n/a		n/a		absent
12.01.2024		n/a		n/a		n/a		absent
13.01.2024		n/a		n/a		n/a		absent
14.01.2024		n/a		n/a		n/a		absent
15.01.2024		n/a		n/a		n/a		absent

16.01.2024	n/a	n/a	n/a	absent
17.01.2024	n/a	n/a	n/a	absent
18.01.2024	n/a	n/a	n/a	absent
19.01.2024	n/a	n/a	n/a	absent
20.01.2024	n/a	n/a	n/a	absent
21.01.2024	n/a	n/a	n/a	absent
22.01.2024	n/a	n/a	n/a	absent
23.01.2024	n/a	n/a	n/a	absent
24.01.2024	n/a	n/a	n/a	absent
25.01.2024	n/a	n/a	n/a	absent
26.01.2024	n/a	n/a	n/a	absent
27.01.2024	n/a	n/a	n/a	absent
28.01.2024	n/a	n/a	n/a	absent
29.01.2024	n/a	n/a	n/a	absent
30.01.2024	n/a	n/a	n/a	absent
31.01.2024	19:20	03:15	07h 55m	fully present

WORKING TIME : 22h 43m				
PRESENT : 03				
ABSENT : 24				
INCOMPLETE : 04				
RECORDS : 10				

Links

- [Arrays](#)
- [IOException](#)
- [Path](#)
- [Scanner](#)
- [StandardCharsets](#)
- [String](#)
- [StringBuilder](#)