

Hinweise für den Prüfling

Bearbeitungszeit (insgesamt): 255 Minuten

Auswahlverfahren

Es gibt zwei Aufgabengruppen A (Algorithmik und objektorientierte Modellierung) und B (Datenbanken und/oder Konzepte und Anwendungen der theoretischen Informatik), aus denen jeweils ein Vorschlag zu bearbeiten ist.

Der vorliegende Vorschlag aus der Gruppe A ist ein Pflichtvorschlag.

Wählen Sie von den zwei vorliegenden Vorschlägen der Gruppe B einen zur Bearbeitung aus.

Der nicht ausgewählte Vorschlag wird 60 Minuten nach Beginn der Bearbeitungszeit von der Aufsicht führenden Lehrkraft eingesammelt.

Erlaubte Hilfsmittel

1. ein Wörterbuch der deutschen Rechtschreibung
2. eine Liste der fachspezifischen Operatoren

Sonstige Hinweise

keine

In jedem Fall vom Prüfling auszufüllen

Name: _____

Vorname: _____

Prüferin/Prüfer: _____

Datum: _____

Fitnessarmband

Fitnessarmbänder erfreuen sich einer immer größeren Beliebtheit. Diese sogenannten Fitnesstracker zeichnen verschiedene Daten auf, um so Rückschlüsse auf den Gesundheitszustand bzw. das sportliche Bewegungsverhalten ziehen zu können. Mit den gewonnenen Daten können Trainingseinheiten gezielt überwacht und ausgewertet werden. Im Folgenden wird ein Ausschnitt eines Fitnesstrackers modelliert.

Innerhalb der Klasse *Fitnessarmband* werden Name, Geschlecht, Körpergröße und Schrittlänge des Benutzers verwaltet. Außerdem erzeugt die Klasse *Fitnessarmband* mittels der Methode *erstelleTagesleistung()* jeden Tag ein Objekt der Klasse *Tagesleistung*, das am Ende des Tages im Feld *meineTagesleistungen* gespeichert wird.

Die Klasse *Tagesleistung* speichert für ein Datum sowohl die Anzahl der Schritte als auch den Puls jeweils zu jeder vollen Minute ($60 \cdot 24 = 1440$ mal) automatisch mit Hilfe der Methode *setzeSchritte(...)* bzw. *setzePuls(...)* in den Feldern *Schritte* bzw. *Pulswerte* ab. Als Methoden existieren *gibMinPuls()*, *gibMaxPuls()* und *gibDurchschnittPuls()*, die den niedrigsten, den höchsten bzw. den durchschnittlichen Tagespuls ermitteln. Die Methode *berechneWegstrecke(double Schrittlänge)* berechnet aus Schrittlänge und den Schritten die zurückgelegte Wegstrecke. Die Methode *setzePuls(int Stunde, int Minute, int Pulswert)* speichert den aktuellen Pulswert am entsprechenden Index im Feld *Pulswerte*.

Aufgaben

- 1.1 Modellieren Sie die Klassen *Fitnessarmband* und *Tagesleistung* gemäß obigen Anforderungen als UML-Klassendiagramm. Auf die Angabe der get/set-Methoden kann verzichtet werden.
(8 BE)
- 1.2 Erläutern Sie anhand des UML-Klassendiagramms aus Aufgabe 1.1 das Geheimnisprinzip.
(5 BE)
- 2.1 Implementieren Sie die Methode *setzePuls(...)* der Klasse *Tagesleistung*.
(3 BE)
- 2.2 Implementieren Sie für die Klasse *Tagesleistung* die Methode *gibDurchschnittPuls()*, die den durchschnittlichen Puls eines vergangenen Tages ermittelt.
(5 BE)
- 3.1 Entwerfen Sie ein Struktogramm für einen Algorithmus *berechneWegstrecke(...)*, der aus der Schrittlänge und den Schritten pro Minute die an einem Tag zurückgelegte Wegstrecke berechnet.
(5 BE)
- 3.2 Implementieren Sie die Methode *bestimmeBesterTag()* der Klasse *Fitnessarmband*, die das Datum liefert, an dem die größte Wegstrecke zurückgelegt wurde.
(7 BE)

- 4 Das Fitnessarmband in der GPS-Version enthält auch einen GPS-Empfänger. Es zeichnet für einen Trainingslauf die aktuelle Position mit Breitengrad, Längengrad und Höhe einmal pro Minute auf. Diese Werte werden mit dem aktuellen Datum in der Klasse *Lauf* (Material 1) gespeichert. Das GPS-Armband kann bis zu 100 Läufe speichern.

- 4.1 Modellieren Sie die Klassen *GPSArmband* und *Lauf* als UML-Klassendiagramm.
Berücksichtigen Sie dabei auch Aufgabe 4.2.

Hinweis

Die Klasse *GPSArmband* enthält auch alle Attribute und Methoden der Klasse *Fitnessarmband* aus Aufgabe 1.

(7 BE)

- 4.2 Ein wichtiger Wert für die Laufanalyse wird mit den Methoden *untersuchen()* und *wasTueIch(...)* der Klasse *Lauf* bestimmt (Material 2).

Analysieren Sie die beiden Methoden und erläutern Sie die Bedeutung des Wertes, den *untersuchen()* liefert.

(13 BE)

- 4.3 Im Feld *Läufe* der Klasse *GPSArmband* sind die Datensätze der einzelnen Läufe nicht zwingend nach Datum sortiert. Für eine differenzierte Analyse der Leistungsentwicklung ist dies aber erforderlich.

Beschreiben Sie einen Algorithmus, der das Feld *Läufe* nach Datum sortiert.

(7 BE)

Material 1**Die Klasse *Lauf***

```
01 public class Lauf {  
02  
03     private doubleint[] Breite;  
04     private doubleint[] Länge;  
05     private int[] Höhe;  
06     private String Datum;  
07     ...  
08 }
```

Material 2**Die Methoden *untersuchen()* und *wasTueIch(...)* der Klasse *Lauf***

```
01 public int untersuchen() {  
02     int Wert = wasTueIch(0, Höhe.length-1);  
03     return Wert;  
04 }  
05  
06 private int wasTueIch(int Untergrenze, int Obergrenze) {  
07     if (Obergrenze == Untergrenze) {  
08         return Höhe[Obergrenze];  
09     }  
10     else {  
11         int Mitte = (Obergrenze + Untergrenze) / 2;  
12         int Wert1 = wasTueIch(Untergrenze, Mitte);  
13         int Wert2 = wasTueIch(Mitte + 1, Obergrenze);  
14         if (Wert1 < Wert2)  
15             return Wert2;  
16         else  
17             return Wert1;  
18     }  
19 }
```