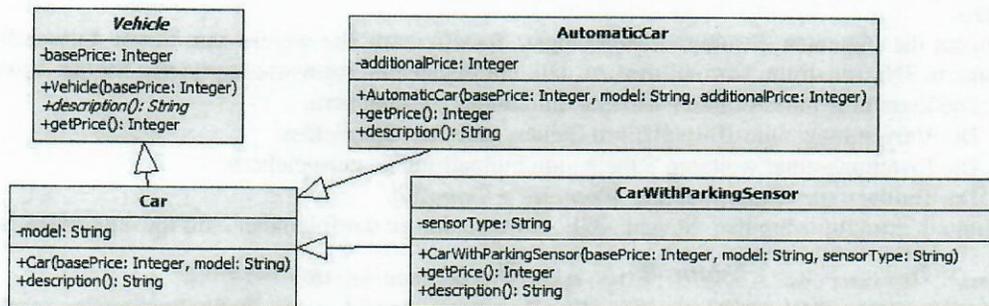


Schriftliche Abschlussprüfung, 2. JULI 2018
 Fachgebiet Informatik in deutscher Sprache
 VARIANTE 1

TEIL 1. Algorithmen und Programmierung

Schreiben Sie ein Programm in Python, C++, Java oder C#, mit den folgenden Bedingungen:

a). **Definiert** die Klassen *Vehicle* (abstrakt), *Car*, *AutomaticCar* und *CarWithParkingSensor* aufgrund des folgenden UML-Diagramms:



- *basePrice* (der Basispreis) und *additionalPrice* (der Mehrpreis) sollen größer als 0 sowie *sensorType* (der Sensortyp) und *model* (das Fahrzeugmodell) sollen nicht leer und echt positiv sein. Die Konstruktoren sollen diese Bedingungen durchsetzen.
 - Die Methode `getPrice()` der Klasse *Vehicle* gibt den Basispreis des Fahrzeuges zurück. Die Methode `getPrice()` der Klasse *AutomaticCar* gibt den Mehrpreis erhöht mit dem Basispreis des Fahrzeuges zurück. Die Methode `getPrice()` der Klasse *CarWithParkingSensor* gibt den Basispreis des Fahrzeuges erhöht mit 2500 zurück.
 - Die Methode `description()` der Klasse *Car* gibt das Fahrzeugmodell zurück. Die Methode `description()` der Klasse *AutomaticCar* gibt die Verkettung vom Text "Automatic car " und dem Fahrzeugmodell. Die Methode `description()` der Klasse *CarWithParkingSensor* gibt die Verkettung vom Text "Car with parking sensor " dem Sensortyp, " " und dem Fahrzeugmodell.
- b). **Eine Funktion definiert**, die eine Liste L der Objekte vom Typ-*Vehicle* bekommt. Die **Funktion** gibt eine Liste von Paare des Typs $\langle \text{Modell}, \text{Fahrzeuganzahl} \rangle$, die für jedes Fahrzeugmodell der Liste L die Anzahl der Fahrzeuge von diesem Modell enthält, zurück.
 - c). **Eine Funktion definiert**, die eine Liste der Objekte vom Typ-*Vehicle* bekommt. Die **Funktion** ordnet mit der Erhaltung der initialen Reihe die Liste um, so dass alle Fahrzeuge, deren Preis im Intervall [1000, 2000] ist, vor den Fahrzeugen, deren Preis kleiner als 1000 oder größer als 4000 ist, erscheinen. Sie dürfen nicht Listen oder weitere Datenstrukturen benutzen.
 - d). **Eine Funktion definiert**, die eine Liste der Objekte vom Typ-*Vehicle* bekommt. Die **Funktion** gibt die Beschreibung aller Fahrzeuge von der Liste auf dem Bildschirm aus.
 - e). **Die Main-Funktion** erstellt eine Liste mit den folgenden Fahrzeugen (wählen Sie die Werte für die un spezifizierte Eigenschaften der Objekte aus): ein Audi, ein Audi mit Automatikgetriebe, ein Toyota, ein Mercedes mit Automatikgetriebe und ein Opel mit Parksensoren. Rufen Sie die Funktion von b). für die bisherige erzeugte Liste auf und geben Sie die zurückgegebene Liste auf dem Bildschirm aus. Rufen Sie die Funktion von c). auf und danach geben Sie mit der Verwendung der Funktion von d). die zurückgegebene Liste auf dem Bildschirm aus.
 - f). Für den benutzten **List**-Typ schreiben Sie eine Spezifikation der benutzten Operationen.

Bemerkung

- **Geben Sie die Programmiersprache an.**
- **Sie dürfen nicht Sorted-Containers oder vordefinierte Sortierfunktionen benutzen.**
- **Andere Funktionen, die nicht in den Aufgaben spezifiziert sind, sollen nicht umgesetzt sein.**

Sie dürfen Datenstrukturen von Standard-Bibliotheken benutzen (Python, C++, Java, C, C#).

TEIL 2. Datenbanken

Gegeben sei eine Datenbank, die Spielergebnisse der aller Spiele der Fußballnationalmannschaft speichert (die Mannschaft darf nur ein Spiel an einem beliebigen Tag spielen). Die Datenbank hat die folgende Struktur:

- Tabelle *Teams* mit Felder: **TId, Country, Continent;**
- Tabelle *Players* mit Felder: **PId, TeamId, Name, Age, NumberOfMatches;**
- Tabelle *Matches* mit Felder: **Team1Id, Team1Goals, Team2Id, Team2Goals, Date, Stadium, NumberOfSeats, City, Country.**

1. Bestimmen Sie die Primäre- und Fremdschlüssel für jede der obigen Tabelle.
2. Bestimmen Sie mindestens vier funktionale Abhängigkeiten, die auf Felder, die nicht Codes repräsentieren, verweisen.
3. Gegeben sei die folgenden Strukturveränderungen. Spezifizieren Sie welche von diesen notwendig sind, um die Datenbank in 3NF (die dritte Normalform) zu sein. Für bejahende Antworte begründen Sie die Auswahl.
 - a. Die Erstellung einer weiteren Tabelle, um Länder zu speichern.
 - b. Die Verwendung eines **DateOfBirth** Feldes, statt des **Age** Feldes.
 - c. Die Erstellung einer weiteren Tabelle, um Fußballstadien zu speichern.
 - d. Das Einfügen der Einschränkung **Team1Id > Team2Id**.
4. Für die initiale Struktur schreiben Sie eine SQL Abfrage, die zu der folgenden Abfrage äquivalent ist:

$$\Pi_{\text{Name}} (\sigma_{\text{Continent} = \text{'Asia'}} (\text{Teams} \otimes_{\text{TId} = \text{TeamId}} \sigma_{\text{NumberOfMatches} > 100} (\text{Players})))$$

5. Schreiben Sie eine SQL Abfrage, die die Gesamtspiellanzahl jeder Fußballnationalmannschaft (**Country, NoMatches**) zurückgibt.

TEIL 3. Betriebssysteme

3.1 Das untenstehende Programmteil wird erfolgreich in der ausführbaren Datei `pr` kompiliert. Das Argument `w` der Funktion `f` spezifiziert, welche FIFO Datei zum Schreiben verwendet werden soll: 0 für `a` und 1 für `b`. Voraussetzend, dass alle Befehle erfolgreich ausgeführt werden, alle notwendigen FIFOs vor jeder Ausführung gelöscht und erneut erstellt werden, und dass `pr` das einzige Programm ist, der auf diese FIFOs zugreift, beantworten Sie die folgenden Fragen.

| | |
|---|---|
| <pre>void f(char* a, char* b, int w, char* s) { int f[2], r=1-w; char c; if(fork() == 0) { f[0] = open(a, w==0 ? O_WRONLY : O_RDONLY); f[1] = open(b, w==1 ? O_WRONLY : O_RDONLY); write(f[w], s, 1); read(f[r], &c, 1); printf("%c\n", c); close(f[0]); close(f[1]); exit(0); } } int main(int n, char** a) { int i; for(i=1; i<n; i+=4) { f(a[i], a[i+1], a[i+2][0]-'0', a[i+3]); } for(i=1; i<n; i+=4) {wait(0);} return 0; }</pre> | <p>a) Zeichnen Sie das Prozesshierarchiediagramm für eine Ausführung, in der <code>pr</code> 4*K Befehlszeilenargumente erhält.</p> <p>b) Was wird die folgende Ausführung drucken? Begründen Sie Ihre Antwort. <code>./pr p q 1 x</code></p> <p>c) Was wird die folgende Ausführung drucken? Begründen Sie Ihre Antwort. <code>./pr p q 1 x p q 0 y</code></p> <p>d) Zeichnen Sie ein Diagramm, das die Kindprozesse und ihre Lese- / Schreibvorgänge in den FIFOs für die Ausführung in (c) darstellt.</p> <p>e) Was wird die folgende Ausführung drucken? Begründen Sie Ihre Antwort. <code>./pr p q 1 x q p 1 y</code></p> |
|---|---|

3.2 Der Befehl `sed s/A/B/` ersetzt die erste Erscheinung aus jeder Zeile des regulären Ausdrucks `A` mit dem String `B` und substituiert jegliche Referenz `\N` aus `B` mit dem Inhalt des `N`-ten ausgeklammerten Ausdrucks aus `A`. Was sind die gedruckten Ergebnisse des folgenden UNIX-Shell-Skripts, wenn sie in einem Verzeichnis ausgeführt werden, das C/C++ - Quelldateien und -Header enthält? Erklären Sie ausführlich die 3. Zeile: die Befehle, die Argumente und die Pipe.

```
1 for F in *.c *.cpp *.h; do
2   if [ -f $F ]; then
3     grep "#include.*" $F | sed "s/^\.*<(.*)>.*$/\1/"
4     fi
5 done | sort
```

BEMERKUNG.

- Alle Prüfungsthemen sind verpflichtend. Bei allen Prüfungsthemen müssen vollständige Lösungen angegeben werden.

BAREM INFORMATICĂ

VARIANTA 1

Subiect 1 (Algoritmica și Programare):

Oficiu – 1p

Definirea clasei abstracte Vehicle – 0.3p din care

atribut – 0.1

metode - 0.2

Definirea clasei Car – 0.3p din care

relația de moștenire – 0.1

atribut – 0.1

metode - 0.1

Definirea clasei AutomaticCar– 1p din care

relația de moștenire – 0.15

atribut – 0.15

constructor – 0.35

metode – 0.35

Definirea clasei CarWithParkingSensor – 1p din care

relația de moștenire – 0.15

atribut – 0.15

constructor – 0.35

metode – 0.35

Funcția de la punctul b) – 2.25p din care

signatura corectă - 0.1p

construire listă perechi conținând modele distincte- 2p

returnare listă rezultat – 0.15

Funcția de la punctul c) – 2.25p din care

signatura corectă - 0.1p

rearanjare listă – 2.15p

Funcția de la punctul d) – 0.4p din care

signatura corectă - 0.1p

afișare descrierii mașini din listă – 0.3p

Funcția principală e) – 0.5p

f) Specificațiile operațiilor folosite pentru tipul de dată Listă– 1p

Subiect 2 (Baze de date)

1. 0.5p (chei primare) + 0.5p (chei externe) = 1p

2. $0.25p \times 4 = 1p$

3. a, c

$2 \times (0.5p \text{ răspuns} + 0.5p \text{ justificare}) = 2p$

4. rezolvarea completă a interogării = 2p

5. rezolvarea completă a interogării = 3p

1p of

Subiect 3 (Sisteme de operare):

3.1.a - diagrama cu un părinte și mulți fii - 0.5p

- K procese fiu - 0.5p

3.1.b - nimic - 0.5p

- se blochează open la deschiderea FIFO-ului - 0.5p

3.1.c - x și y - 0.5p

- în ordine nedeterminabilă - 0.5p

3.1.d - diagrama circulară cu două procese și două FIFO-uri - 1p

3.1.e - nimic - 0.5p

- se blochează open-urile și creează deadlock - 0.5p

3.2 Rezultate - lista sortată a fișierelor header sistem incluse în surse - 1p

3.2 Linia 3 - grep: extrage liniile care includ fișiere header sistem - 0.5p

- grep: detaliere expresie regulată - 0.5p

- pipe: transmite output-ul lui grep ca input lui sed - 0.5p

- sed: păstrează pe linie doar numele fișierului header - 1p

- sed: detaliere comandă și expresie regulată - 0.5p