BABEŞ-BOLYAI UNIVERSITÄT CLUJ-NAPOCA
FAKULTÄT FÜR MATEMATHIK UND INFORMATIK

Schriftliche Abschlussprüfung, 4. September 2019
Fachgebiet Informatik in deutscher Sprache

VARIANTE 1

BEMERKUNG.

- Alle Prüfungsthemen sind verpflichtend. Bei allen Prüfungsthemen müssen vollständige Lösungen angegeben werden.
- Die Mindestnote für das Bestehen der Abschlussprüfung ist 5,00.
- Die Arbeitszeit beträgt 3 Stunden.

TEIL Betriebssysteme

1 Beantworten Sie die folgenden Fragen, berücksichtigend, dass alle erforderlichen Header-Dateien im folgenden Programm enthalten sind und dass alle Befehle erfolgreich ausgeführt werden.

```
1  int n=0, a[2], b[2];
2  void w(int p[2], char c) {
3    n++;
4    if(fork() == 0) {
5      close(p[0]); write(p[1], &c, 1); close(p[1]);
6      exit(0);
7    }
8  }
9  void r(int p[2]) {
10   char c;
11   n++;
12   if(fork() == 0) {
13     close(p[1]);
14     if(read(p[0], &c, 1) > 0) {printf("%c\n", c);}
15     close(p[0]);
16     exit(0);
17   }
18 }
19 int main(int argc, char** argv) {
20   pipe(a); pipe(b);
21   w(a, 'x'); w(a, 'y'); r(a); r(a);
22   close(a[0]);close(a[1]);close(b[0]);close(b[1]);
23   for(int i=0; i<n; i++) {wait(0);}
24   printf("%d\n", n);
25   return 0;
26 }
```

a) Zeichnen Sie die Hierarchie der erzeugten Prozesse, einschließend das Vater Prozess.

b) Was stellt von der Durchführung des Programms dar?

c) Was stellt von der Durchführung des Programms dar, wenn Zeile 21 durch den folgenden Code ersetzt wird?

```
w(a, 'x'); w(b, 'y'); r(a); r(b);
```

d) Was stellt von der Durchführung des Programms dar, wenn Zeile 21 durch den folgenden Code ersetzt wird?

```
w(a, 'x'); w(a, 'y'); r(a); r(b);
```

e) Erklären und begründen Sie die Funktionsweise des Programms, wenn Zeile 21 wie unter Punkt d) ersetzt wird, und zusätzlich Zeile 22 gestrichen wird.

2 Beantworten Sie die folgenden Fragen zum unterstehenden Shell UNIX-Skript.

```
1  #!/bin/bash
2  for F in *.sh; do
3    A=`grep "^[ \t]*[^ \t#]" $F | wc -l`
4    B=`wc -l < $F`
5    if [ $A -lt `expr $B - $A` ]; then
6      echo $F
7    fi
8  done
```

a) Was stellt von der Durchführung des Skript dar?

b) Was stellt von der Durchführung des Skript dar, wenn er in einem Verzeichnis ausgeführt wird, das nur das Skript selbst enthält?

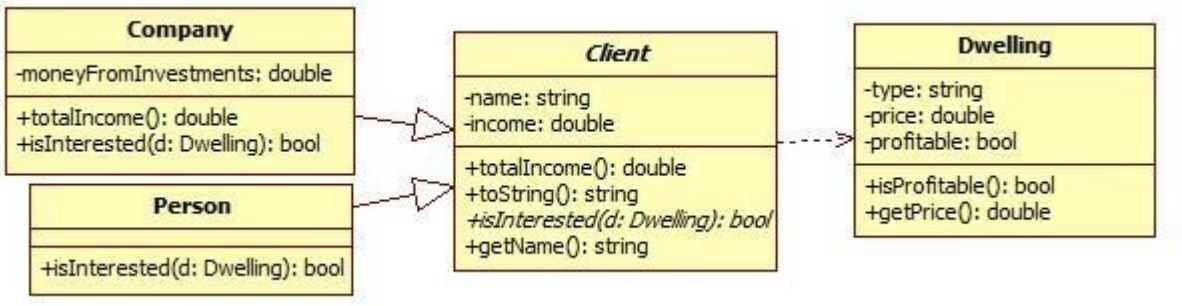
c) Erklären Sie den regulären Ausdruck in Zeile 3 ausführlich

VARIANTE 1

TEIL Algorithmen und Programmierung

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

a) Definiert die Klassen **Client** (Kunde), **Person (**Person**)**, **Company** (Unternehmen), **Dwelling (**Behausung**)** aufgrund des folgenden UML-Diagramms (die Konstruktoren sind nicht im Diagramm enthalten). Nur die Methoden von b) sollen implementiert werden, der Rest sollen nur deklariert werden.



- das Attribut *name* der Klasse **Client** (der Kundenname) soll mindestens drei Zeichen haben und *income* (das Einkommen) soll echt positiv sein. Die Konstruktoren sollen diese Bedingungen durchsetzen.
- die abstrakte Klasse **Client** stellt eine abstrakte Methode **isInterested** bereit.
- Das gesamte Einkommen eines Unternehmens ist dessen Einkommen (*income*) zusammen mit der Summe, die von Kapitalanlagen (*moneyFromInvestments*) geleistet wird. Das gesamte Einkommen eines anderen Kundes ist sein Einkommen.
- Eine Person hat Interesse für eine Behausung (**Dwelling**) wenn der Preis der Behausung durch 360 geteilt kleiner als die Hälfte des gesamten Einkommens des Persons ist. Ein Unternehmen hat Interesse für eine Behausung, wenn der Preis der Behausung durch 12 geteilt kleiner als das gesamte Einkommen des Unternehmens ist und wenn die Behausung profitable ist.
- die Methode **toString** gibt für Personen und auch Unternehmen den Name (*name*) erweitert mit dem gesamten Einkommen zurück.

b) Implementiert die folgenden Methoden aus dem Diagramm von a): die Konstruktoren der Klassen **Client**, **Person**, **Company**, **Dwelling**; die Methoden **totalIncome** der Klassen **Client** und **Company** und die Methoden **isInterested** der Klassen **Person** und **Company**.

c) Definiert eine Funktion, welche eine Liste von Kunden zusammen mit einer Ordnungsrelation (dargestellt als eine boolean Funktion, die zwei Kunden vergleicht) als Parameter bekommt. Die Funktion sortiert die Liste der Kunden nach der Reihenfolge der Ordnungsrelation. Das Sortieren wird mit einem Algorithmus, dessen Komplexität $\theta(n \cdot log_2 n)$ ist, gemacht werden.

d) Definiert eine Funktion, welche ein assoziatives Feld (Dictionary) von Kunden (Schlüssel - Kundenname, Wert - der Kunde) und eine Behausung als Parameter bekommt. Die Funktion gibt mit Hilfe der Funktion von c) eine alphabetisch nach Name sortierte Liste mit allen Kunden, die Interesse für die Behausung haben, zurück.

e) Erzeugt in der Main-Funktion ein assoziatives Feld (Dictionary) von Kunden und fügt die folgenden Kunden hinzu (wählen Sie die Werte für die unspezifizierten Eigenschaften der Objekte aus): zwei Unternehmen, eine mit dem Namen "TwoStar" und die andere mit dem Namen "Fort" und zwei Personen, eine mit dem Namen "Ana" und die andere mit dem Namen "Mihai". Erzeugen Sie zwei Behausungen: eine vom Typ "Wohnung", die 150000 kostet und nicht profitable ist; und eine vom Typ "Haus", die 500000 kostet und profitable ist. Für das assoziative Feld der Kunden und jede Behausung rufen sie die Funktion von d) auf. Geben Sie die Kunden der entstehenden Listen mit der **toString** Methode auf dem Bildschirm aus.

f) Für den **Dictionary**-Typ schreiben Sie eine Spezifikation der Operationen **Hinzufügen** und **Suchen**.

Bemerkung

- **Geben Sie die Programmiersprache an.**
- **Sie dürfen keine weiteren Methoden definieren, außer Konstruktoren und den Methoden, die im UML-Diagramm dargestellt sind.**
- **Sie dürfen nicht Sorted-Containers oder vordefinierte Sortierfunktionen benutzen.**

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

VARIANTE 1

TEIL Datenbanken

Gegeben sei eine Datenbank für die elektronische Stimmabgabe bei einem Referendum. Die Datenbank hat die folgende Struktur:

- Tabelle ***Persons*** mit den Feldern: **PId**, **Name**, **CNP**, **CityId**, **CountyId**, **RegionId, Voted;**
- Tabelle ***Votes*** mit den Feldern: **VId**, **Date, Time;**
- Tabelle ***Answers*** mit den Feldern: **VId**, **QId**, **YesAnswer, NoAnswer, Cancelled**;
- Tabelle ***Questions*** mit den Feldern: **QId**, **Text, CategoryName**, **CategoryDescription.**

Eine Antwort zu einer Frage gilt als ungültig (**cancelled**) falls weder *Ja*, noch *Nein* angekreuzt wurde oder falls beide angekreuzt wurden. Die Felder **YesAnswer, NoAnswer, Cancelled** und **Voted** können nur die Werte 0 oder 1 haben.

1. Bestimmen Sie die Kandidat- und die Fremdschlüssel für jede der obigen Tabellen.
2. Bestimmen Sie mindestens vier funktionale Abhängigkeiten auf Feldern, die nicht Kandidatschlüssel repräsentieren.
3. Gegeben seien die folgenden Strukturveränderungen. Bestimmen Sie welche von diesen notwendig sind, um die Datenbank in 3NF (die dritte Normalform) zu sein. Für bejahende Antworten begründen Sie die Auswahl.
   a. Die Erstellung einer weiteren Tabelle, um Kategorien zu speichern.
   b. Die Beseitigung des Feldes **Cancelled** aus der Tabelle ***Answers*.**
   c. Die Verwendung in der Tabelle ***Votes*** eines einzigen Feldes, das sowohl das Datum als auch die Zeit der Stimmabgabe speichert.
   d. Das Einfügen eines neuen Feldes **PId** in der Tabelle ***Votes***, das auf Tupeln aus der Tabelle ***Persons*** verweist.
4. Für die gegebene Struktur, schreiben Sie eine SQL Abfrage, die zu der folgenden Abfrage äquivalent ist:

$$\Pi_{\text{Text, Time}} (\sigma_{\text{Time} >18:00} (\textbf{\textit{Votes}} \otimes_{\text{Votes.VId=Answers.VId}} \textbf{\textit{Answers}} \otimes_{\text{Questions.QId=Answers.QId}} \sigma_{\text{CategoryName='EU'}} (\textbf{\textit{Questions}})))$$

5. Für die gegebene Struktur, schreiben Sie eine SQL Abfrage, die für jede Frage die gesamte Anzahl von Antworten, die gesamte Anzahl von gültigen „Ja" Antworten, die gesamte Anzahl von gültigen „Nein" Antworten und die gesamte Anzahl von ungültigen Antworten angibt (***Text, NumberAnswers, NumberYes, NumberNo, NumberCancelled***).

BAREM INFORMATICĂ
VARIANTA 1

Subiect (Algoritmică şi Programare)

Oficiu – 1p
Definirea clasei abstracte Client – 0.6p din care
- atribute – 0.1
- constructor si metode – 0.5

Definirea clasei Company– 1.5p din care
- relația de moștenire – 0.2
- atribut – 0.1
- constructor – 0.4
- metode – 0.8

Definirea clasei Person – 0.7p din care
- relația de moștenire – 0.2
- constructor – 0.2
- metoda – 0.3

Definirea clasei Dwelling – 0.2p din care
- atribute– 0.1
- constructor - 0.1

Funcția de la punctul c) – 3p din care
- signatura corectă - 0.4p
- sortare in $\theta(n \cdot log_2 n)$ - 2.5p
- returnare listă rezultat – 0.1

Funcția de la punctul d) – 1.5p din care
- signatura corectă - 0.1p
- construire lista clienti in ordine alfabetica dupa nume – 1.3p
- returnare listă rezultat – 0.1

Funcția principală e) – 0.5p
f) Specificaţiile operaţiilor **adăugare** și **căutare** pentru tipul de dată **Dicționar** – 1p

Subiect Baze de date

1. 0.5p (chei candidat) + 0.5p (chei externe) = 1p
2. 0.25p x 4 = 1p
3. a, b
   - 2 x (0.5p răspuns + 0.5p justificare) = 2p
4. rezolvarea completă a interogării = 2p
5. rezolvarea completă a interogării = 3p

1p of

Subiect Sisteme de operare

Oficiu – 1p

1.a Diagramă cu procesul părinte având patru procese fiu – 1p
1.b Va afișa “x y 4” sau “y x 4” – 1p
1.c Identic cu punctul 3.1.b – 1p
1.d Va afișa “x 4” sau “y 4” – 1p
1.e Va afișa “x” sau “y” și se va bloca la linia 14 pentru că pipe-ul este deschis pentru scriere – 1.5p

2.a Numele scripturilor Shell din directorul curent cu mai puține linii de cod decât restul liniilor din script – 1p
2.b Nu va afișa nimic – 1p
2.c Inceput de linie, 0 sau mai multe spații sau taburi, orice caracter care nu e spațiu, tab sau diez – 1.5p