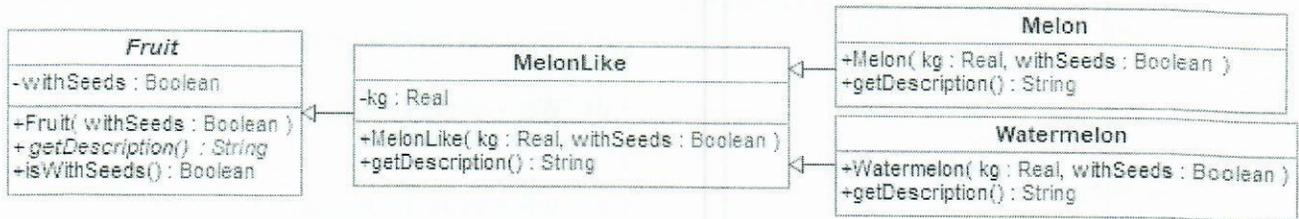


Schriftliche Abschlussprüfung, 3. Juli 2017
 Fachgebiet Informatik in deutscher Sprache
 VARIANTE 2

TEIL 1. Algorithmen und Programmierung

Schreiben Sie ein Programm in Python, C++, Java oder C#, welches:

- a). Die Klassen **Fruit** (Obst), **MelonLike** (Melone), **Melon** (Honigmelone) und **Watermelon** (Wassermelone) basiert auf das folgende UML Diagramm definiert



- *kg* soll größer als 0 sein. Die Konstruktoren sollen diese Bedingungen durchsetzen.
 - Die abstrakte Klasse **Fruit** soll eine abstrakte Methode *getDescription()* (Beschreibung) enthalten.
 - Hat ein Obst Samen, gibt die Methode *isWithSeeds()* der Klasse **Fruit** true, andernfalls false zurück. Die Methode *getDescription()* der Klasse **MelonLike** gibt eine Zeichenkette zurück. Die Zeichenkette setzt sich durch die Verkettung von: *die Masse*, dem Text „melonlike“ und dem Text „with seeds“ (falls das Objekt Samen hat) oder „without seeds“ (falls das Objekt keine Samen hat) zusammen. Die Methoden *getDescription()* der Klassen **Watermelon** und **Melon** geben den Rückgabewert der Basisklasse erweitert mit „melon“ bzw. „watermelon“ zurück.
- b). **Eine Funktion definiert**, die den möglichen Einfügungsindex eines **Fruit** in einer alphabetischen, nach dem Wert der *getDescription()* Methode sortierten Liste von **Fruit** zurückgibt. Bitten verwenden Sie die binäre Suche.
- c). **Eine Funktion definiert**, die mit der Verwendung der Methode von b) ein **Fruit** in einer alphabetischen sortierten Liste von **Fruit** einfügt.
- d). **Eine Funktion definiert**, die einen booleschen Wert *withSeed* und eine Liste von **Fruit** bekommt. Die Funktion gibt Obst mit/ohne Samen von dem Wert des Parameter *withSeeds* abhängig auf dem Bildschirm aus.
- e). **Die Main-Funktion** erstellt eine Liste von Obst: **Watermelon** ohne Samen von 6 kg, **Melon** mit Samen von 10 kg, **MelonLike** ohne Samen von 11 kg, **Watermelon** mit Samen von 13 kg. Mit der Verwendung der Funktion von c) fügen Sie ein **Watermelon** ohne Samen von 12 kg in der zuvor erstellten Liste ein. Geben Sie, unter Verwendung der Funktion von d), Obst mit und ohne Samen getrennt auf dem Bildschirm aus.
- f). Für den benutzten **List**-Typ schreiben Sie eine Spezifikation der benutzten Operationen.

Bemerkung

- Die ausgewählte Programmiersprache wird angegeben.
- 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).

TEIL 2. Datenbanken

- a. Erstellen Sie eine relationale Datenbank mit allen Tabellen in 3NF, um folgende Informationen über TIFF zu verwalten:
- **Standorte:** Standort-Id, Name, Adresse.
 - **Filme:** Titel, Jahr, Liste von Genre (jedes Genre hat einen Id, Name und Beschreibung), Liste von Schauspieler (jeder Schauspieler hat einen Id und einen Namen), Liste von Filmprojektionen (für jede Projektion wird ein Filmprojektions-Id, einen Standort-Id, Datum und Zeit angegeben).
 - **verkaufte Eintrittskarten:** Filmprojektions-Id, Reihe und Platznummer.

Begründen Sie, dass die erhaltene Datenbank in 3NF ist durch Identifizierung der funktionalen Abhängigkeiten.

- i. Die Liste der Standorte (Name, Adresse), wo mindestens eine *Komödie* und mindestens ein *Drama* projiziert worden sind.
- ii. Die gesamte Anzahl der verkauften Eintrittskarten, die für Filme in denen *Alain Delon* spielt und die in der *Piata Unirii* projiziert worden sind.
- iii. Die Liste der Filme (Titel, Jahr) für die die größte Anzahl von Eintrittskarten verkauft worden sind.

TEIL 3. Betriebssysteme

3.1 Voraussetzend, dass alle Befehle aus dem untenstehenden Programmteil erfolgreich durchgeführt worden sind, beantworten Sie die folgenden Fragen:

<pre> 1 int main(){ 2 int p[2], i=0; 3 char c, s[20]; 4 pipe(p); 5 if (fork()==0){ 6 close(p[1]); 7 while(read(p[0], &c, sizeof(char))){ 8 if(i < 5 i > 8){ 9 printf("%c", c); 10 } 11 i++; 12 } 13 printf("\n"); close(p[0]); 14 exit(0); 15 } 16 printf("Result: \n"); 17 strcpy(s, "exam not passed"); 18 close(p[0]); 19 write(p[1], s, strlen(s)*sizeof(char)); 20 close(p[1]); 21 wait(NULL); 22 return 0; 23 }</pre>	<ol style="list-style-type: none"> a) Zeichnen Sie die Hierarchie der erzeugten Prozesse, inklusive des Vater Prozess. b) Geben Sie jede vom Programm angezeigte Zeile, mitsamt dem Prozess welcher diese Zeile ausdrückt. c) Wie viele Zeichen werden aus dem pipe gelesen? d) Wie wird die Terminierung der Prozesse beeinflusst, falls die Zeile 20 fehlt? e) Wie wird die Terminierung der Prozesse beeinflusst, falls die Zeilen 20 und 21 fehlen?
---	--

3.2 Angenommen das untenstehende Shell UNIX Skript wird durchgeführt, beantworten Sie die folgenden Fragen.

<pre> 1 f=`find . -type f` 2 d=`find . -type d` 3 4 for x in \$f; do 5 for y in \$d; do 6 if [\$x = \$y]; then 7 echo "OK" 8 fi 9 done 10 done</pre>	<ol style="list-style-type: none"> a) Wie oft wird „OK“ angezeigt? Begründen Sie Ihre Antwort. b) Welchen Wert nimmt die Variable f an? c) Welchen Wert nimmt die Variable d an? d) Welche Werte nimmt die Variable x an? e) Welche Werte nimmt die Variable y an?
--	---

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.

Subiect 1 (Algoritmă și Programare):

Oficiu – 1p

Definirea claselor *Fruit* și *MelonLike* – 1p din care

relația de moștenire – 0.25

atribute – 0.25

constructor – 0.25

metode - 0.25

Definirea claselor *Watermelon* și *Melon* – 1p din care

relația de moștenire – 0.25

constructor – 0.25

metode – 0.5

Funcția de la punctul b) – 2p din care

signatura corectă - 0.1p

algoritmul de căutare binară - 1.8p

returnare rezultat - 0.1p

Funcția de la punctul c) - 2p din care

signatura corectă - 0.1p

determinare poziție de inserare - cu fct de la b) – 0.2p

inserarea elementului pe poziția determinată anterior – 1.7p

Funcția de la punctul d) – 1p din care

signatura corectă - 0.1p

parcurgere listă – 0.4p

verificare condiție – 0.1p

accesare și tipărire element – 0.4p

Funcția principală e) – 0.5p

f) Specificațiile operațiilor folosite pentru tipul de dată *Listă* – 1.5p**Subiect 2 (Baze de date)****1 punct** oficiu

Problema a:

2 puncte pentru tabelele în 3NF**2 puncte** pentru justificare:**1 punct** definiții**1 punct** explicații

Problema b:

1.5 puncte pentru i**1 punct** pentru ii**2.5 puncte** pentru iii**Subiect 3 (Sisteme de operare):**

Oficiu – 1p

3.1 – 5p din care

a) Diagrama ierarhiei - 1p

b) Linia părintelui – 0.5p

Linia fiului – 0.5p

c) 15 caractere – 1p

d) Niciun proces nu se termină. Fiul blocat la read, părintele la wait – 1p

e) Procesele se termină, pipe-ul fiind închis la terminarea părintelui – 1p

3.2 – 4p din care

a) Nu se afișează nimic – 1p

Justificare – 1p

b) Numele și calea tuturor fișierelor normale din directorul curent și toate subdirectoarele – 0.5p

c) Numele și calea tuturor directoarelor din directorul curent și toate subdirectoarele – 0.5p

d) Numele și calea fiecărui fișier lista din \$f – 0.5p

e) Numele și calea fiecărui director din lista \$d – 0.5p