

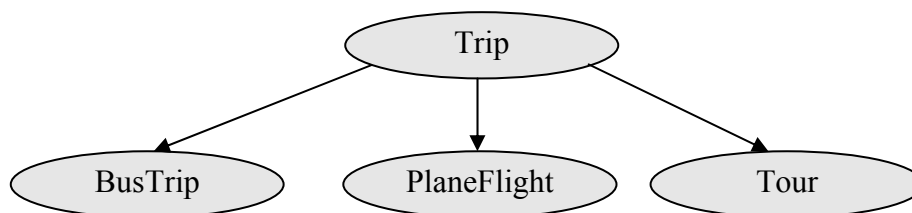
**СОФИЙСКИ УНИВЕРСИТЕТ  
“СВ. КЛИМЕНТ ОХРИДСКИ”****ФАКУЛТЕТ ПО МАТЕМАТИКА  
И ИНФОРМАТИКА****ДЪРЖАВЕН ИЗПИТ  
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАЦИОННИ СИСТЕМИ”****ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)  
08.09.2010 г.****Време за работа – 3 часа**

Драги абсолвенти, спазвайте стриктно следните указания:

- Пишете само на предоставените ви листове без да ги разкопчавате;
- Попълнете факултетния си номер горе вдясно на всички страници;
- Решението на всяка от задачите се разполага само в пространството от края на условието на тази задача до началото на условието на следващата задача (могат да се използват и двете страни на листата).

Изпитната комисия ви пожелава успешна работа.

**Задача 1 (30 т.)** Да се реализира следната йерархия от класове:



**Trip** е абстрактен клас, описващ пътуване, който поддържа следните свойства (properties):

- *крайна дестинация* (низ с произволна дължина);
- *описание* (низ с произволна дължина, само за четене);
- *дължина в километри* (цяло число);
- *цена* (число с плаваща запетая).

**BusTrip** е клас, описващ пътуване с автобус. Класът поддържа следното допълнително свойство:

*фирма превозвач* (низ с произволна дължина).

Описанието на пътуване с автобус трябва да е от следния вид:

„Пътуване <дължина> км до <дестинация> на цена <цена> лв. с автобус на фирма <превозвач>“

**PlaneFlight** е клас, описващ пътуване със самолет. Класът поддържа следното допълнително свойство:

*код на полет* (последователност от две букви и до 5 цифри, напр. LH3434)

Описанието на полет със самолет трябва да е от следния вид:

„Пътуване <дължина> км до <дестинация> на цена <цена> лв. с полет <код>“

**Tour** е клас, описващ екскурзия като последователност от пътувания. Класът поддържа следните допълнителни свойства:

*пътуване с пореден номер i* (указател към обект от клас Trip).

Дължината и цената на екскурзията са сумите на стойностите на съответните свойства за отделните пътувания. Описанието на екскурзията се състои от описанията на всички пътувания и завършва с обобщаваща информация:

„Обща дължина <дължина> км на цена <цена> лв.“

За горните класове да се реализират:

1. „голямата четворка“ — конструктор по подразбиране, конструктор за копиране, оператор за присвояване, деструктор, за класовете, в които са необходими;
2. подходящи селектори и мутатори за свойствата *дестинация*, *дължина*, *цена*, *фирма превозвач*, *код на полет* и *пътуване с пореден номер i*;
3. селектори за свойствата *описание* за отделните класове.

За класа **Tour** да се реализират следните член-функции:

4. *sliceTour(char const\* start, char const\* end)*, която построява нова екскурзия, включваща само пътуванията от дестинация *start* до дестинация *end* включително. Внимание: пътуването с крайна дестинация *start* не се включва, но пътуването, което непосредствено го следва, трябва да се включи. При невалидни дестинации да се построи празна екскурзия.
5. *priceReport()*, която отпечатва всички пътувания в екскурзията, подредени в намаляващ ред по цена за 1 км.



**Задача 2 (5 т.)** Да се дефинира на езика Хаскел функцията **sumProd**, която по даден списък от списъци от цели числа **ll** да намира сумата от произведенията на елементите на тези списъци в **ll**, които не съдържат отрицателни числа.

```
sumProd :: [[Integer]] -> Integer
```

```
sumProd ll = .....
```

**Задача 3 (15 т.)** Дадени са следните дефиниции на типове на езика Хаскел, описващи:

Пътуване (**Trip**) като наредена тройка от крайна дестинация, дължина в километри и цена

Екскурзия (**Tour**) като списък от пътувания

```
type Trip = (String, Integer, Float)
```

```
type Tour = [Trip]
```

Да се дефинират функции:

- **discount**, която по дадена екскурзия намира нова екскурзия, в която цената на всяко пътуване с дължина над **len** км е намалена с 10%.

```
discount :: Tour -> Integer -> Tour
```

```
discount tour len = .....
```

- **shortenTour**, която по дадена екскурзия намира нова екскурзия, в която всяка последователност от пътувания, започваща от дестинация **from** и завършваща в дестинация **to** е заменена с единственото пътуване **trip**. *Внимание:* последователност, отговаряща на описанието започва непосредствено след пътуване, чиято **крайна** дестинация е **from**, но не го включва!

```
shortenTour :: Tour -> String -> String -> Trip -> Tour
```

```
shortenTour tour from to trip = .....
```

**Задача 4 (15 т.)** Информационна система обслужва електронна резервация и продажба на билети за автобуси. Системата позволява резервация на билет, отмяна на резервация, покупка на билет, отмяна на пътуване (например при затваряне на даден път през зимата).

- А. Да се опише модел на потребителските случаи за системата. Да се определят актьорите, границите на системата, евентуални връзки с други системи, възможно най-подробен списък на потенциални потребителски случаи. Да се начертае диаграма за този модел.
- В. Да се състави диаграма на състоянията за един билет. Да се обяснят преходите и условията за тях.



**Задача 5 (10 т.)**

Дадена е базата от данни Movies.

Таблицата **Studio** съдържа информация за филмови студиа:

name – име, първичен ключ;

address – адрес.

Таблицата **Movie** съдържа информация за филми. Колоните *title* и *year* заедно формират първичния ключ.

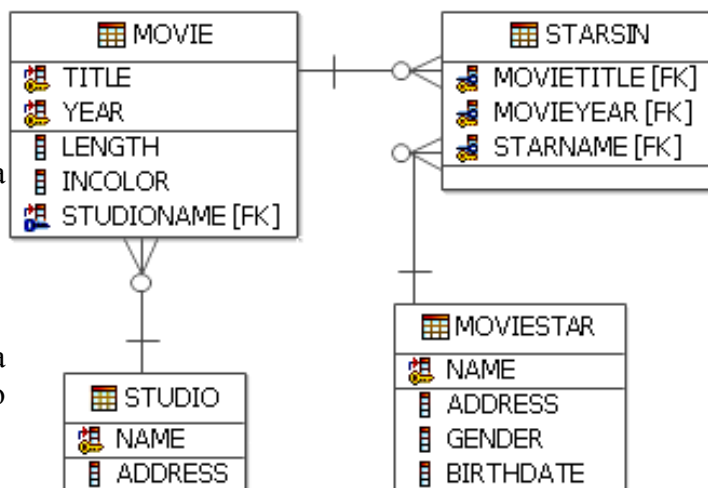
title – заглавие;

year – година, в която филмът е заснет;

length – дължина в минути;

incolor – 'Y' за цветен филм и 'N' за черно-бял;

studioName – име на студио, външен ключ.



Таблицата **MovieStar** съдържа информация за филмови звезди:

name – име;

address – адрес;

gender – пол, 'M' за мъж и 'F' за жена;

birthdate – рождена дата.

Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите колони заедно формират първичния ключ. Колоните *movietitle* и *movieyear* образуват външен ключ:

movietitle – заглавие на филма;

movieyear – година на заснемане на филма;

starname – име на филмовата звезда, външен ключ.

1. Напишете заявка, която извежда имената на всички актриси, които не са играли в нито един филм в периода 1970-1979 г. Имената на актрисите, за чиито участия няма информация в базата от данни, също да трябва да бъдат изведени.



2. Напишете заявка, която за всяка филмова звезда, родена преди 1990 г. (без значение от какъв пол), извежда информация за това, на каква възраст е играла за първи път във филм. Не трябва да бъдат изведени имената на звезди, за чиито участия във филми няма информация в базата от данни. Резултатът трябва да има следния вид:

name	debut_age

**Задача 6 (15 т.)** Информационна система за видеоклипове съхранява информация за видеоклипове и потребителите, които ги изпращат.

За всеки потребител задължително се съхранява следната информация: уникален номер, уникално потребителско име и парола.

За всеки видеоклип задължително се съхранява следната информация: уникален номер, автор (потребител от системата), заглавие, дължина в секунди. Ако е налична, следната информация също се съхранява: производител на камерата, с която е сниман клипът, както и модел на камерата.

Всеки потребител може да притежава много клипове, но един клип не може да бъде споделен между двама потребители.

Всеки потребител може да се абонира за един или повече други потребители на системата. Даден потребител не може да се абонира два пъти за един и същ потребител.

а) Създайте E/R модел на БД, която съхранява гореописаната информация. Обосновете модела. Начертайте E/R диаграма на модела.

б) Преобразувайте E/R диаграмата към релационни схеми. Премахнете излишествата, където това е възможно.

**Задача 7 (10 т.)** Дадена е релацията **R(ABCDEF)** с функционални зависимости (ФЗ):

**AC  $\rightarrow$  B, BD  $\rightarrow$  F, F  $\rightarrow$  CE.**

**а)** Да се намерят всички възможни ключове за **R**.

**б)** Да се посочи ФЗ, която нарушава НФБК. Като се използва тази ФЗ, да се декомпозира **R** на две релации **R1** и **R2**. В коя нормална форма (НФ) е **R1**? В коя НФ е **R2**?

**в)** Ако декомпозираме **R** в релацията **S(ACDE)**, посочете една нетривиална ФЗ от релацията **R**, която е в сила и за релацията **S**.