

**СОФИЙСКИ УНИВЕРСИТЕТ  
„СВ. КЛИМЕНТ ОХРИДСКИ“**



**ФАКУЛТЕТ ПО МАТЕМАТИКА  
И ИНФОРМАТИКА**

**ДЪРЖАВЕН ИЗПИТ  
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАЦИОННИ СИСТЕМИ”**

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)  
15.07.2014 г.**

**Време за работа – 3 часа**

*Драги абсолвенти:*

- Попълнете факултетния си номер в горния десен ъгъл на всички страници;
- Пишете само на предоставените листове без да ги разкопчавате;
- За всяка от задачите, беловата с решението може да е само на листите, на които е изписано условието на съответната задача, или на празна страница след условието. При необходимост пренасяте решението на подпечатан нов лист със заглавен текст „Задача N, стр. M, ф.н. F“, където M ( $M \geq 1$ ) е поредния номер допълнителен лист за задача N, а F е вашият факултетен номер.

*Изпитната комисия ви пожелава успешна работа!*

**Задача 1.** (10 т.) Задачата да се реши на езика C++ или Java. В началото на вашето решение посочете кой език сте избрали.

Дадена е квадратна матрица от цели числа с размери  $10 \times 10$ , която описва лабиринт. Стойност 0 в дадена клетка означава „стена“, а стойност 1 означава „проходима клетка“. Даден е символен низ, съдържащ само буквите E, W, N и S, които указват едностъпкови придвижвания в съответните географски посоки: N – нагоре, E – надясно, S – надолу, W – наляво.

Да се напише функция `walk`, която получава матрица и символен низ от вида, определен по-горе и проверява дали символният низ задава валиден път започващ от някоя проходима клетка на лабиринта, състоящ се само от проходими клетки и завършващ в долния десен ъгъл на лабиринта. Функцията да връща булева стойност – *истина*, ако такава клетка има и даденият низ задава валиден път и *лъжа* в противен случай.

**Задача 2.** (10 т.) *Задачата да се реши на езика C++ или Java. В началото на вашето решение посочете кой език сте избрали.*

А) Да се дефинира структура `ChessPosition`, описваща коректна позиция на фигура върху шахматна дъска (координатите на позицията са от 'А' до 'Н' по едното измерение и от 1 до 8 по другото).

Да се дефинира абстрактен клас (или интерфейс) `ChessPiece`, описващ шахматна фигура със следните операции:

- `ChessPosition getPosition()` – Връща позицията на фигурата на дъската;
- [подходящ тип] `allowedMoves()` – Връща списък (колекция) с всички възможни позиции, до които дадена фигура може да достигне с един ход;
- [булев тип] `captures(ChessPosition pos)` – Проверява дали фигурата “владее” позицията `pos`, подадена като параметър, т.е. дали позицията е в списъка с възможните ходове на фигурата. Булевият тип да бъде булевият тип в езика, който сте избрали (напр. `bool`, ако пишете на C++).

Б) Да се дефинират класовете `Rook` и `Knight` – наследници на `ChessPiece`, описващи съответно шахматните фигури топ и кон.

В) „Стабилна конфигурация“ наричаме такава подредба на фигурите върху дъската, при която никоя фигура да не е върху позволен ход на друга фигура (т.е. никои две фигури да не се „бият“). Да се дефинира функцията `allMoves ([подходящ тип] pieces[, ...])`, която за списъка (колекцията) `pieces`, съдържащ произволен брой разнородни шахматни фигури, отпечатва на конзолата всеки възможен ход на фигура от `pieces` такъв, че след изпълнението му списъкът с фигури да описва стабилна конфигурация. Информацията за ходовете да съдържа типа на фигурата, старата позиция и новата позиция, например:

`Rook A1 -> B1`

`Knight B3 -> A5`

*Забележка: Реализирайте всички конструктори и други операции, които смятате, че са необходими на съответните класове.*

**Задача 3.** (10 т.) *Задачата да се реши на езика Scheme или Haskell. В началото на вашето решение посочете кой език сте избрали.*

А) Напишете функция `totalMin`, която за списък от едноместни числови функции връща тази функция  $f$  от списъка, за която  $f(0)$  е минимално.

Б) Напишете функция `chainMinCompositions`, която получава като аргумент едноместна числова функция  $f$  и генерира безкрайния поток (за Хаскел – безкрайния списък)  $F_0, F_1, F_2, \dots$ , където:

$$F_0 = id$$

$$F_1 = f$$

$$F_i = F_{i-1} \circ F_{i-2}, \text{ ако } i > 1 \text{ и } F_{i-1}(j) \neq F_{i-2}(j), \text{ за някое цяло число } j \in [0, i]$$

$$F_i = totalMin \{F_0, F_1, \dots, F_{i-1}\}, \text{ ако } i > 1 \text{ и } F_{i-1}(j) = F_{i-2}(j), \text{ за всяко цяло число } j \in [0, i]$$

*Забележка: с  $id$  е означена функцията „идентитет“, като  $id(x) = x$  за произволно  $x$ , а с  $f \circ g$  е означена композицията на функциите на  $f$  и  $g$ , като  $(f \circ g)(x) = f(g(x))$ .*

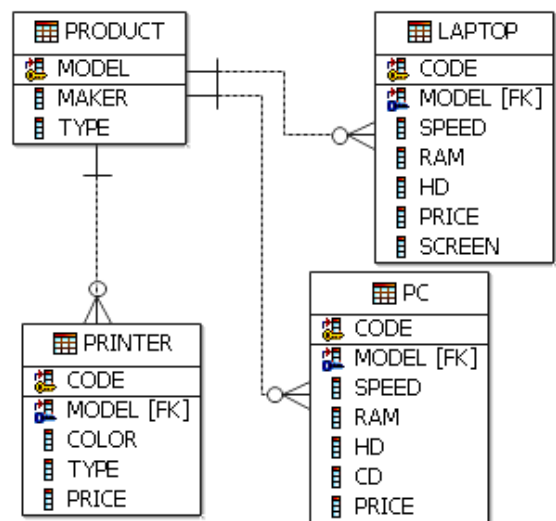
**Задача 4.** (10 т.) Дадена е базата от данни РС. В нея се съхранява информация за три вида продукти – настолни компютри, лаптопи и принтери.

Таблицата **Product** съдържа базова информация за всеки продукт:

- *model* – модел на продукта, първичен ключ;
- *maker* – производител на продукта;
- *type* – един от следните типове: 'PC', 'Laptop' или 'Printer'.

Таблицата **PC** съдържа специфична информация за настолните компютри:

- *code* – уникален идентификатор на дадена компютърна конфигурация, първичен ключ;
- *model* – модел на настолния компютър, външен ключ към Product.model. Може да имаме няколко различни компютърни конфигурации от един и същ модел, но с различни параметри;
- *speed* – тактова честота на процесора в MHz;
- *ram* – количество RAM памет в MB;
- *hd* – размер на твърдия диск в GB;
- *cd* – скорост на CD устройството;
- *price* – цена на настолния компютър.



Таблицата **Laptop** съдържа специфична информация за лаптопите. Атрибутите са аналогични на тези на PC, но липсва атрибутът CD и има атрибут за размера на екрана.

Таблицата **Printer** съдържа информация за принтерите:

- *code*, *model*, *price* – аналогични на едноименните атрибути в PC;
- *color* – 'y' за цветен принтер, 'n' за черно-бял;
- *type* – тип на принтера – 'Laser', 'Jet', 'Matrix'.

За така описаната база данни решете следните задачи:

1. Напишете заявка, която извежда всички производители на настолни компютри, които произвеждат и лаптопи.
2. Напишете заявка, която извежда кодовете, моделите и размерите на екраните на всички лаптопи, чиито производители имат не повече от три модела принтери (евентуално 0).

**Задача 5.** (10 т.) Моделирайте база от данни за Световното първенство по футбол през 2014 г. Базата от данни трябва да съдържа следната информация:

**Групи (Groups)**

- Име на група (gname) - низ, точно един символ (например А, В и т.н.), уникален идентификатор за всяка група
- Първо място в групата (fplace) – името на държавата (отбора) спечелила първо място в групата, низ
- Второ място в групата (splace) – името на държавата (отбора) спечелила второ място в групата, низ

**Отбори (Teams)**

- Държава (country) – уникален идентификатор за всеки отбор, низ
- Брой участия на световни първенства (num\_ws), цяло число

**Играчи (Players)**

- Име на играч (pname) – уникален идентификатор за всеки играч, низ
- Номер на фланелка (num\_player), цяло число
- Брой отбелязани голове на това световно първенство (num\_goals), цяло число

**Мачове (Matches)**

- Номер на мач (mnumber) – уникален идентификатор за всеки мач, число
- Вид на фаза (tstage) - от коя фаза е мача (групова фаза, елиминационна фаза), низ
- Фаза от директната елиминация (stage) – низ, например осминафинали, четвъртфинали, полуфинали и т.н. Може да бъде NULL ако мача е от групова фаза.
- Резултат от мача (result) – низ, точно 1 символ (например 1 при победа на първия отбор, 2 при победа на втория отбор)
- Голова разлика (scores) – низ, точно 5 символа
- Дата на провеждане на мача (date) – дата
- Час на провеждане на мача (time) – час

В сила са следните ограничения:

- В една група има много (четири) отбора, а всеки отбор е част само от една група.
- За един отбор може да играят много играчи (до 22-ма), а даден играч може да играе само за един отбор.
- Във всеки един от мачовете могат да играят един срещу друг точно два отбора, като единият условно е домакин, а другият гост.

Направете E/R модел на описаната по-горе база от данни. Отбележете първичните ключове. Преобразувайте направения от вас E/R модел в релационен модел.

**Задача 6.** (10 т.) Прочетете описанието на информационната система от задача 9. След това решете дадените по-долу задачи.

- А) Да се изготви потребителски случай (пълно описание), който описва как даден потребител използва системата, за да провери дали за дадена дата и час има свободна подходяща зала (като вид и капацитет) и след това запазва залата за провеждане на някакво събитие.
- Б) В потребителския случай включете поне два алтернативни сценария.
- В) Напишете нефункционални изисквания към потребителския случай.
- Г) Начертайте диаграма на състояния (State Machine), която описва състоянията на една заявка за зала от описаната в задача 9 система. Диаграмата да включва подходящи етикети върху дъгите свързващи състоянията, които описват логиката (преходи и условия), по която се осъществява преходът от едно в друго състояние. Ако е нужно, напишете кратки разяснения към диаграмата.

**Задача 7.** (10 т.) Да се направи информационен модел на електронен бележник с адреси и телефони на хора, който да замести хартиения бележник. Да се аргументира изборът за всеки обект от модела, структурата на модела и броя на появяванията на обектите. Бележникът съдържа: собствено, бащино и фамилно име на роднини, колеги и познати, адрес, съдържащ пощенски код, името на града, име и номер на улица/булевард, дата на раждане за обаждане за поздрав за рожден ден, телефони (ако има) – домашни и служебни. Това съдържание е записано във XML документ, в който има информация за нула, един или повече души. Да се състави описание на документния му тип с използване на DTD или XML Schema. Да се създадат примерен XML документ, в който има данни за поне двама души, и съответен DTD или XML Schema документ.

**Задача 8.** (10 т.) „209А“ е кодовото име на проект на система за запазване на зали във ФМИ. Системата трябва да има уеб-базиран интерфейс и да позволява изпращане на заявки за зали към администрацията от всеки преподавател. В заявката могат да се указват тип зала, брой места, наличие на техника, ден, час, продължителност и дали заявката е за еднократно или повтарящо се използване, както и за коя дисциплина, специалност, курс и поток се запазва залата. Системата трябва да може да се интегрира с базата данни на ФМИ, която съдържа потребителски профили на преподаватели, както и списък на специалности, дисциплини, курсове и потоци. „209А“ трябва да има и административен интерфейс, който да позволява одобряване и отхвърляне на заявки, както и създаване, редактиране и изтриване на информация за зали. Трябва също да е възможно да бъде предлаган списък от зали отговарящи на дадена заявка и генериране на план на заетите зали за даден период време.

Да се предложи примерна структура на работните пакети (work breakdown structure) за проекта, с не повече от 4 нива и не повече от 20 възела.