КОНСПЕКТ

ИНФОРМАТИКА

Важни правила!!!

8,9,10 клас

I. Алгоритми. Основни видове алгоритми.
II. Информация. Основни единици за измерване на информацията.
III. Бройни системи(двоична, осмична, шестнайсетична). Всички начини за преминаване от една бройна система в друга(чрез деление с остатък, рекурсивно, чрез методи в Java)
IV. Логически функции.
V. Java. Структура на програмата. – (част за импорт, заглавна част, част за декларации на глобални променливи, част за декларации на конструктори, методи, главен метод). Синтаксис на езика.
VI. Променливи в Java. Константи. Статични променливи. Достъп до променливите(public, private, protected)
VII. Типове данни в Java
    1. Примитивни - (числови(byte, int,long; float, double); текстови(char); логически(boolean))
    2. Структурни – String, Integer, Double, Character.
VII. Оператори в Java
    1. Оператор за присвояване(=)
    2. Оператор за въвеждане на данни от клавиатурата(Scanner)
    3. Оператор за извеждане на данни (System.out.print/***ln/f***())
    4. Условен оператор – if(condition){true}else if {}…..else if()…else….
    5. Оператор за многовариантен избор

    ```
    switch(variable){
    case value1: cons1;break;
    case value2: cons2;break;
    ……..
    default:  consn;
    ```

    6. Цикли. Оператори за цикъл. Условия за начало и край.

    а) increment/decrement- i++;++I;i--;--i

    б) автоматичен цикъл **for(int i=0;i<n;i++){тяло на цикъла}**

в) цикъл с предусловие – **while(cond) {тяло на цикъла}**

г) цикъл с постусловие – **do {тяло на цикъла} while (cond)**

д) вложени цикли

е) важни циклични алгоритми(sum, min, max и др.)

7. Методи в Java(return;void). Синтаксис.
8. Вградени функции (class Math(Math.max(x, y); Math.sqrt(y); Math.pow(x, y); Math.random();Math.round(x), Дата, време и др., Math.ceil(8.234), Math.round(33.456), Math.floor(4.765))
9. Рекурсия.
10. String.

****

11 клас

VIII. Структури от данни.

1. Статични структури. Масиви

а) **едномерни масиви** – синтаксис, създаване на масив, въвеждане на елементи в масив, извеждане на елементи от масив. Обхождане на масив.

Основни алгоритми с едномерни масиви.

б) **двумерни масиви** – деклариране, основни действия с двумерни масиви(сума на главен диагонал, второстепенен диагонал)

2. Динамични структури. Колекции

а) **Списъци** – Интерфейс List(ArrayList, LinkedList, Stack).

Деклариране. Основни методи при работа със списъци(add, remove, insert elements). Обхождане на списъци. Извеждане на данни от списъка. Прилагане на основните алгоритми.

б) Опашки – Интерфейс Queue(ArrayDeque, PriorityQueue)

в) Множества/уникални стойности/ - Интерфейс Set(HashSet, TreeSet)

Интерфейс Map(HashMap, TreeMap)

IX. ООП.

1. Структура на ООП програма – полета, конструктор, методи.
2. Наследяване. „has a”, “is”.
3. Полиморфизъм. Предефиниране, претоварване на методи. Upcasting;

4. Абстрактен клас. Интерфейс.
5. Comparable; Comparator;
6. Изключения в Java.

X. Други теми: enum, регулярни изрази.

****

12 клас

XI. Релационни БД – MySql; Ms Access

a) Mysql – Основни понятия(таблица, заявка, ред(запис, кортеж), колона(поле, атрибут), релации)
b) Типове данни- числови, текстови, дата и време
c) Основни операции – създаване на БД(create database); създаване на таблица(create table), модифициране на таблица(alter table); задаване на първичен ключ(primary key), външен ключ(foreign key); изтриване на таблица(drop table), изтриване на БД(drop database)
d) Допълнителни свойства на полетата – autoincrement, default….
e) Попълване на таблица(insert into <table> values),
f) Заявки
   - Заявки за извличане на данни –
     - ✔ Цялата информация
     - ✔ По критерий – WHERE
     - ✔ Изчисляващи справки(!!!по редове)
     - ✔ Логически функции(if, case)?
     - ✔ Агрегатни функции(!!!по колони)- Sum(), avg(),min(), max(),count()
     - ✔ Групиране на данни – Group by column1,column2
     - ✔ Сортиране на данни – Sort by column1,column2
     - ✔ Клауза Having
   - Многотаблични заявки
     - ✔ Join, Left Join, Right Join
   - Заявки за действия – променят данните в таблиците
     - ✔ Update data
     - ✔ Delete … - изтриване на редове по критерий
     - ✔ Функции за работа с текст
     - ✔ Потребителски функции и процедури?

XII. Създаване на приложения

1. Свързване на Java&MySql

2. Създаване на „statement“: Четене на БД, заявки в БД.

XIII. Графичен модул в Java ?

Познаване на основните елементи(Label, TextBox, RadioButton, Button, ComboBox)

Задачи за подготовка

Книга Наков1

Книга Наков2 - подробна

1. Задачи с условен оператор, цикли - СофтУни(Снукър, Спирт, Екскурзия,делители, прости числа,шахматна дъска)
2. Задачи от отделяне на цифри - число - най- голямо число;
3. Задачи с бройни системи -
4. Комбинации -
5. **Обработка на стрингове**.(пеещи думи; **прав, обратен ред на дума в изречение**)
6. Еднакви триъгълници
7. Максимална редица от еднакви елементи
8. Трионообразна редица
9. Двоично-примитивни числа
10. Сортировки
11. Търсене в масив - линейно, двоично
12. Сложност на алгоритми; Cheat Sheet;

Data Structure Complexity Chart

Data Structures	Space Complexity	Average Case Time Complexity			
		Access	Search	Insertion	Deletion
Array	O(n)	O(1)	O(n)	O(n)	O(n)
Stack	O(n)	O(n)	O(n)	O(1)	O(1)
Queue	O(n)	O(n)	O(n)	O(1)	O(1)
Singly Linked List	O(n)	O(n)	O(n)	O(1)	O(1)
Doubly Linked List	O(n)	O(n)	O(n)	O(1)	O(1)
Hash Table	O(n)	N/A	O(1)	O(1)	O(1)
Binary Search Tree	O(n)	O(log n)	O(log n)	O(log n)	O(log n)

Search Algorithms

Search Algorithms	Space Complexity	Time Complexity		
		Best Case	Average Case	Worst Case
Linear Search	O(1)	O(1)	O(n)	O(n)
Binary Search	O(1)	O(1)	O(log n)	O(log n)

Sorting Algorithms

Sorting Algorithms	Space Complexity	Time Complexity		
		Best Case	Average Case	Worst Case
Selection Sort	O(1)	O(n^2)	O(n^2)	O(n^2)
Insertion Sort	O(1)	O(n)	O(n^2)	O(n^2)
Bubble Sort	O(1)	O(n)	O(n^2)	O(n^2)
Quick Sort	O(log n)	O(log n)	O(n log n)	O(n log n)
Merge Sort	O(n)	O(n)	O(n log n)	O(n log n)
Heap Sort	O(1)	O(1)	O(n log n)	O(n log n)

13. Двумерни масиви- деклариране, инициализция, въвеждане на данни, отпечатване,

    главен диагонал, второстепенен диагонал, сума на елементите, сума по колони, сума по редове, сума по диагонали, макс ел, мин ел,(всички, по редове, по колони)

14. Задача:

    Даден е стринг. Създайте програма, която проверява дали всяка дума от стринга може да се въведе в ред на матрица.

    Ако да, да се добави и да се изведе матрицата, обърната обратно

    ВХОД:

    Варна, София, Ловеч, Котел ->

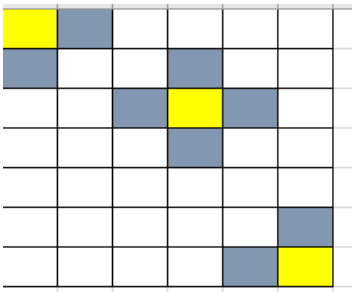
В	а	р	н	а
С	о	ф	и	я
Л	о	в	е	ч
К	о	т	е	л

а	н	р	а	В
я	и	ф	о	С
ч	е	в	о	Л
л	е	т	о	К

Варна, София, Ловеч, Пловдив -”Не може да се въведе в матрица”

****

15. Задача: Максимална сума от съседни клетки в двумерен масив



Рекурсия

Задачи от рекурсия

1) Извеждане на поредица от числа в прав ред пр. **n=5 ; Изход: 12345**
2) Извеждане на поредица от числа в обратен ред . пр. **n=5; Изход: 54321**
3) Сума на първите **n**- числа
4) Факториел - произведението на първите **n** числа
5) Бройни системи

$$\sqrt{1+\sqrt{2}+\sqrt{3}+\cdots\ldots\ldots+\sqrt{n}}\,,n\geq 1$$

6)

```java
public static void recFunc(int i)
{
    if(i % 10 != 0)
    {
        System.out.println(i);
        recFunc(i / 10);
    }
}
```

Запишете какво ще се изведе на екрана след обръщението RecFunc(1234) (C#) / recFunc(1234) (Java).

7)

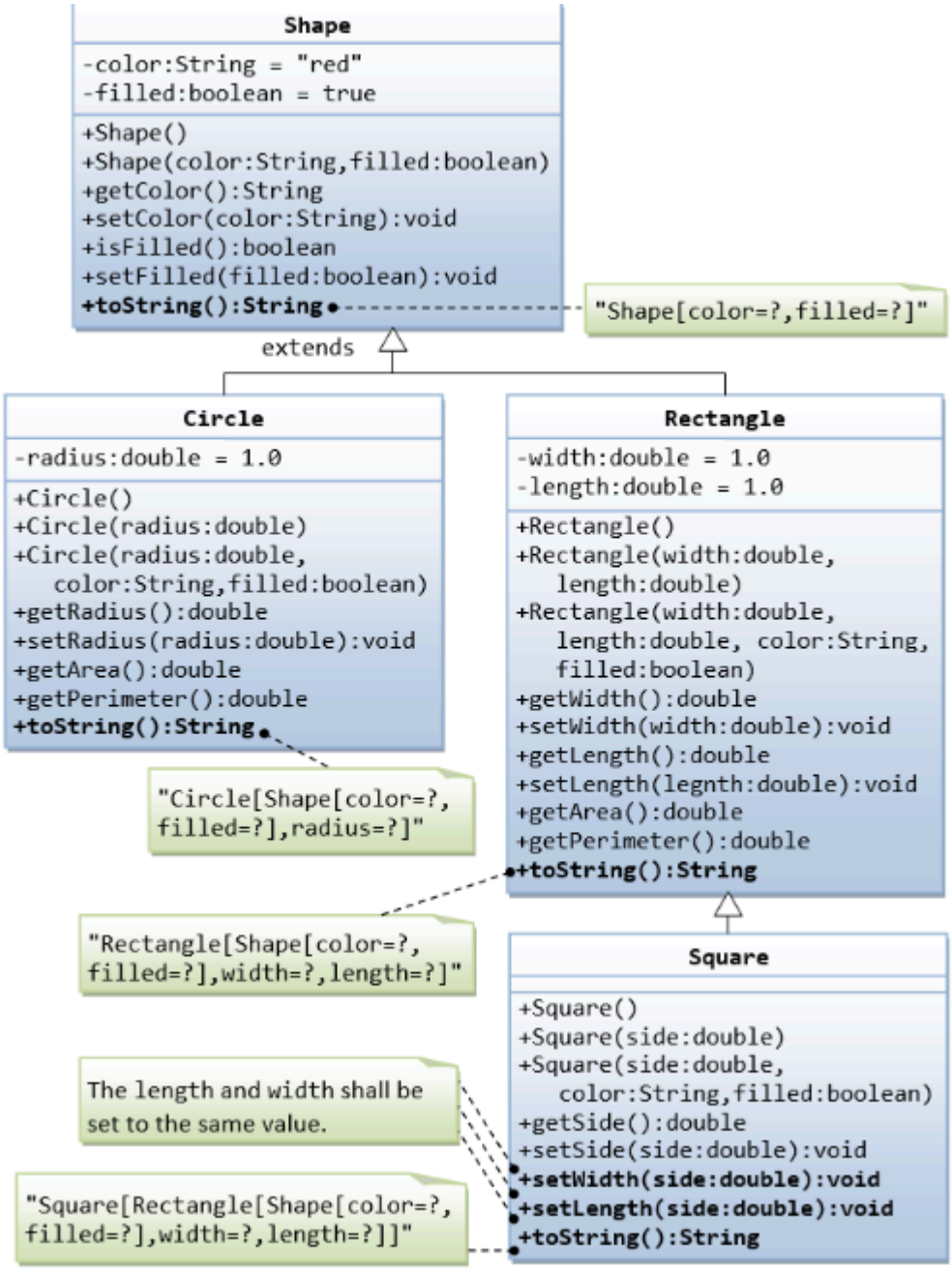
```java
public static void main(String[] args) {
    counter(3);
}
private static void counter(int n){
    if (n == 0)
        return;
    System.out.print(n);
    counter(n - 1);
}
```

8) А) 3210; Б) 1230; В) 123; Г) 321.

ООП

Задача 1

Наследяване на класове



Пробна матура 1 - [Тест](#) говори

Zada

[Задачи](#)

MySQL

Абстрактни класове

Задача за 31.03.2013г.

Дефинирайте клас ***Fighter***, който представя боец със следните характеристики:

- **name: String; до 30 символа**
- **age- int; до 20 години**
- **height - double >1.4м**
- **weight - int;>40кг**
- **abstract Power();**
- **toString();**

****

Дефинирайте клас ***Judist***, който наследява ***Fighter.***

Класът има допълнително поле ***belt u*** реализира метода ***Power()***.

- **belt - int(бял- 1; жълт - 2; оранжев - 3; зелен - 4; син- 5; кафяв - 6; черен -7;**
- **Power() - (height*100+weight)*belt**
- **предефинирайте toString();**

****

Дефинирайте клас ***Boxer***, който наследява ***Fighter.***

Класът има допълнително поле ***strength*** и предефинира метода ***Power()***.

- **strength - 1..7**
- **Power() - 10*strength*(weight/(height*height))**
- **предефинирайте toString();**

****

Създайте тестов клас Test.

Създайте два отбора - TEAMJ и TEAMB; Всеки отбор да има по 5 състезателя.

Сортирайте отборите низходящо по Power(), а при еднакви сили - съответно по **belt** или **strength**;

****

Задача

Създайте клас Point - int x, int y;

Създайте клас LIne (Point A, Point B) - A - начало на отсечката, B- край на отсечката

Създайте метод за намиране дължината на отсечката.

Създайте клас Circle(Point center, int r)

Създайте тестов клас Test.

Създайте следните обекти:

p1->Point(-5,12);

p2->Point(4,-8);

c1-> Circle((2,5),4)

l1-> Line((-7,5),(3,-8)); - Изведете (A,B, d())- d()- дължина на отс.

****

Задача Cars

****

Задача 3. Зъболекар (15)

Доктор Вадѝм Моларов е зъболекар. В дневника му всеки от пациентите му е записан с име и фамилия. За всеки зъб на пациент доктор Моларов поддържа текстов запис, в който записва проблемите на зъба (празен низ, ако зъбът е здрав). Напишете програма **dentist**, която:

- Дава възможност за въвеждане на данните от дневника в компютър чрез използване на стандартния вход.
- Намира средния брой на проблемните зъби, които пациентите имат, и извежда номериран списък на всички пациенти (име_и_фамилия, интервал, брой_проблемни_зъби), които имат повече проблеми от този среден брой. Списъкът да е подреден намаляващо по броя проблемни зъби, а при еднакъв брой – по прав азбучен ред на фамилните имена на пациентите.

ПРИМЕР

ИЗГЛЕД НА ВХОДНИЯ ДИАЛОГ

```
Брой пациенти: 27
Пациент #1
Име и фамилия: Драган Пенев
Зъб #1: кариес
Зъб #2:
Зъб #3:
Зъб #4: гангрена
Зъб #5: за вадене
…
Зъб #32: кариес
Пациент #2
…
```

ИЗГЛЕД НА ИЗХОДА

```
1. Драган Пенев 6
2. Минка Желева 5
3. Ана Якимова 5
```

Пробна матура 1 Отговори

зад25

Пробна матура 2 Отговори

Матура май/2022

Скала за оценяване

Двоично търсене -итерация

Матура - август 2022

Пробна матура 3

Тест Sql(MySql)

а.
```
static void Main(string[] args)
{ int a = 3;
  int b;
  b = a++;
  System.Console.WriteLine(a);
  System.Console.WriteLine(b);
}
```

б.
```
static void Main(string[] args)
{ int a = 3;
  int b;
  b = ++a;
  System.Console.WriteLine(a);
  System.Console.WriteLine(b);
}
```

в.
```
static void Main(string[] args)
{ int a = 3;
  int b = 5;
  a--;
  b++;
  b=a++ + b;
  System.Console.WriteLine(a);
  System.Console.WriteLine(b);
}
```

г.
```
static void Main(string[] args)
{ int a = 3;
  int b = 5;
  a--;
  ++b;
  a = a + b;
  System.Console.WriteLine(a);
  System.Console.WriteLine(b);
}
```

д.
```
static void Main(string[] args)
{ int a = 0, c = 12;
  int b = 0, d = 5;
  a = c / d;
  b = a + b;
  System.Console.WriteLine(a);
  System.Console.WriteLine(b);
}
```

е.
```
static void Main(string[] args)
{ int a = 0, c = 12;
  int b = 10, d = 5;
  a = c % d;
  b = b / a;
  System.Console.WriteLine(a);
  System.Console.WriteLine(b);
}
```

Задачи за упражнение

1 част - АЛГОРИТМИЧНИ ЗАДАЧИ

1. На **n** картончета, подредени в редица са написани последователно числата от 1 до **n**, след което картончетата за разместени и са подредени в нова редица. Напишете програма, която намира в разбъркания ред най- голямото и най- малкото число на тези картончета, което е останало на мястото си.

   ВХОД: На първи ред е **n** - броят на картончетата, а на следващия - разбърканата радица.

   ИЗХОД: Да се изведат двете числа от условието. Ако няма такива или е само едно - извеждаме две нули.

Вход: 8 3 **2** 1 5 8 4 **7** 6	Вход: 8 3 1 2 5 8 4 **7** 6	Вход: 8 3 2 1 5 8 4 6 7
Изход: 2 7	Изход: 0 0	Изход: 0 0

2. ***Едно цяло положително число е съвършено, ако е равно на сумата от всички свои делители, без себе си.***

   Вход: - **n** - цяло число; ***пр. 1000***

   Изход: Всички *съвършени* числа<n***;пр. 6 28 496***

3. Да с***е табулира функцията за интервала [m,n] - въвеждат се от клавиатурата.(x,m,n - цели числа)***

f(x)=	x^2 - 5, 0<x
	1, 0<=x<=1
	x^2+2x+3, x>1

4. Алгоритъм на Лун за валидиране номер на кредитна карта

4561 2612 1234 5467

Стъпка 1: Умножете всяка от цифрите на нечетни позиции*2

8_12_ 4_2_ 2_6_ 10_ 12_

Стъпка 2: Ако някое от числата е>=10, извадете от него 9;

8_3_ 4_2_ 2_6_ 1_ 9_

Стъпка 3: Заместете с новите цифри:

8531 4622 2264 1437

Стъпка 4:

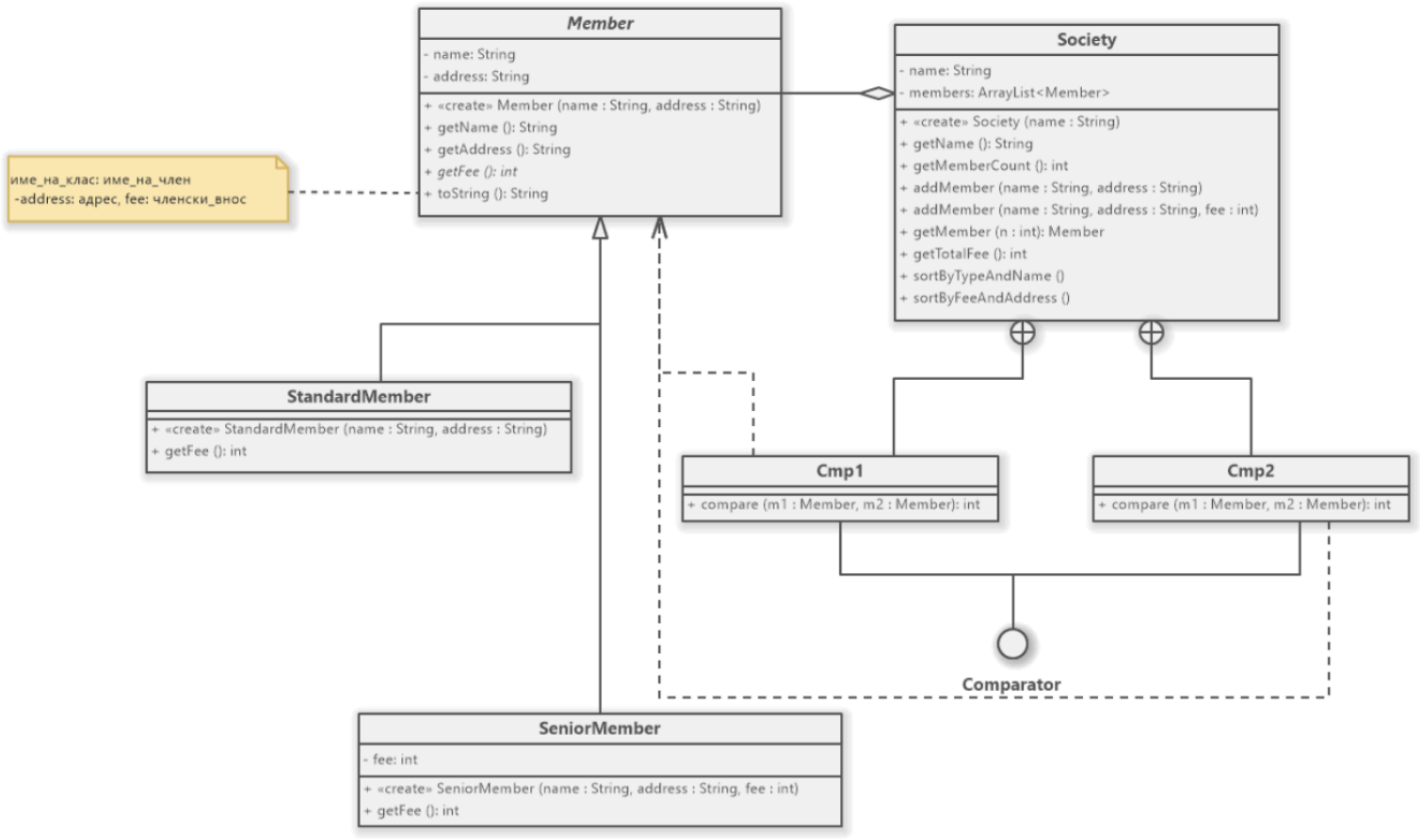
Съберете всички цифри.8+5+3+1+4+6+2+2+2+2+6+4+1+4+3+7=60

Ако полученото число е кратно на 10 => кодът в валиден

2 ЧАСТ - ООП

1. Кухня

2. Задача **Клуб** (15 т.) файл Club.java
Любителски клуб отговаря на следната UML клас-диаграма:



- ☐ Абстрактният клас Member има свойства name (име на член) и address (негов адрес).
- ☐ В класа се декларира и абстрактен метод *getFee*(), който връща членския внос, събиран от съответния член.
- ☐ Дефинира се методът toString(), който връща низ със следната структура: името на класа, двоеточие, интервал, името на члена, нов ред, интервал, тире, думата address, двоеточие, интервал, адрес на члена, запетая, интервал, думата fee, двоеточие, интервал, стойност на членския внос. Наследници на Member са класовете StandardMember и SeniorMember. Членският внос на стандартен член е фиксиран на 30 долара.

- ☐ Класът Society има свойствата name (име на клуба) и members (динамичен масив от членове).
- ☐ Класовете Cmp1 и Cmp2 служат за сортировъчните методи по описаните по-долу приоритети.
- ☐ Методът getTotalFee връща общия членски внос за клуба.

Реализирайте описаната йерархична структура.

Във файл society.txt са записани данни със следната структура:

- Ред 1: име на клуба
- Всеки от останалите редове може да описва член на клуба, данните за който са разделени с точка и запетая и са в следната последователност: име на човека, адрес и, ако е старши член – неговия членски внос. Редове, които се отклоняват от този формат, се игнорират.

Изведете на стандартния изход списък 1:

- първи ред – името на клуба, интервал, думите total fee, знак = и общия членски внос на клуба;
- всеки два от останалите редове описват един член във формата, определен в метода toString() на класа Member.

Членовете в списък 1 трябва да са подредени така: първо старшите, а после стандартните, а при еднакво старшинство – по азбучен ред на имената. Въведете от стандартния вход цяло число *n*.

Изведете на стандартния изход списък 2 – членовете, чийто членски внос не е по-малък от *n*. Подредбата в списък 2 трябва да е намаляващо по членския внос, а при еднакъв членски внос – по азбучен ред на адреса.

Примерен входен файл society.txt

```
Elite
John Hopkins;Washington, Blue Street 9;77 Mery
Dey;Humbert City, Mainstreet 121
Very important:
Bishop Tailor;New York, 77 str., 2110;57
Allen Pitty;Soldoo village
Pippy Poland;Boston, Pouring str., 137;passed away
Petty Lenon;Pardington, Flowers str,12
Ain Moore;Priston, Missy Lane, 122
Mister Tsukimoto;Tokyo, Sinzuku, 65-21;78;traveling home
Poppy Deals;Bourger town, Pee str., 77;57
```

Изходен списък 1

```
Elite total fee=311
SeniorMember: Bishop Tailor
 -address: New York, 77 str., 2110, fee: 57
SeniorMember: John Hopkins
 -address: Washington, Blue Street 9, fee: 77
```

```
SeniorMember: Poppy Deals
 -address: Bourger town, Pee str., 77, fee: 57
StandardMember: Ain Moore
 -address: Priston, Missy Lane, 122, fee: 30
StandardMember: Allen Pitty
 -address: Soldoo village, fee: 30
StandardMember: Mery Dey
 -address: Humbert City, Mainstreet 121, fee: 30
StandardMember: Petty Lenon
 -address: Pardington, Flowers str,12, fee: 30
```

Примерен вход за списък 2

```
50
```

Изходен списък 2

```
SeniorMember: John Hopkins
 -address: Washington, Blue Street 9, fee: 77
SeniorMember: Poppy Deals
 -address: Bourger town, Pee str., 77, fee: 57
SeniorMember: Bishop Tailor
 -address: New York, 77 str., 2110, fee: 57
**********************************************************************************
```

3.

Create a class called 'Matrix' containing constructor that initializes the number of rows and number of columns of a new Matrix object. The Matrix class has the following information:

1.number of rows of matrix

2 number of columns of matrix

3 - elements of matrix in the form of 2D array

The Matrix class has methods for each of the following:

1 - get the number of rows

2 - get the number of columns

3 - set the elements of the matrix at given position (i,j)

4 - adding two matrices. If the matrices are not addable, "Matrices cannot be added" will be displayed.

5 - multiplying the two matrices

4.

Create a class named Pizza that stores information about a single pizza. It should contain the following:

- Private instance variables to store the size of the pizza (either small, medium, or large), the number of cheese toppings, the number of pepperoni toppings, and the number of ham toppings.

- Constructor(s) that set all of the instance variables.
- Public methods to get and set the instance variables.
- A public method named calcCost( ) that returns a double that is the cost of the pizza.
  Pizza cost is determined by:
  Small: $1tt + $2 per topping
  Medium: $12 + $2 per topping
  Large: $14 + $2 per topping
- public method named getDescription( ) that returns a String containing the pizza size, quantity of each topping.

Write test code to create several pizzas and output their descriptions. For example, a large pizza with one cheese, one pepperoni and two ham toppings should cost a total of $22. Now Create a PizzaOrder class that allows up to three pizzas to be saved in an order. Each pizza saved should be a Pizza object. Create a method calcTotal() that returns the cost of order. In the runner order two pizzas and return the total cost.

5.

3 ЧАСТ - БАЗИ ДАННИ - MySql

Задача

Справочник

4 ЧАСТ - СТРУКТУРИ ОТ ДАННИ

1. Съставете метод с параметър целочислен масив и връща същият масив, като местата на елементите са разменени симетрично спрямо средния - първият става последен, вторият предпоследен и т.н
2. Съставете метод, който има за параметър два стринга и връща резултат колко пъти вторият стринг се среща в първия.

3. Съставете метод с параметър стринг. Методът да връща стойност *true/false* - съответно ако стинга е симетричен (първи=последен, втори=предпоследен…) или не е.(abba - true; abab - false;)
4. Даден е едномерен масив B с n реални числа. Съставете метод, който има за параметър масива B и извежда подмасив от всички елементи на B, които са по- големи от средноаритметичното от съседните два.

   ВХОД: 6

   3 5 2 4 7 3

   ИЗХОД: 5 7

ПРИМЕРНИ ЗАДАЧИ!!!

```
 CREATE DATABASE MARKET12GG;

USE MARKET12GG


CREATE TABLE BRANDS

(

ID SMALLINT AUTO_INCREMENT NOT NULL,

NAME CHAR(30),

MARKET_SHARE DOUBLE,

PRIMARY KEY(ID));


****

CREATE TABLE CAMERAS

(

ID SMALLINT PRIMARY KEY AUTO_INCREMENT,

MODEL CHAR(10),

BRAND_ID SMALLINT,
```

```
MPIX INT,
ZOOM INT,
PRICE INT,
FOREIGN KEY(BRAND_ID) REFERENCES BRANDS(ID)
ON DELETE CASCADE
ON UPDATE CASCADE);

****
INSERT INTO BRANDS VALUES
(1,'BESTCAM','0.35'),
(2,'AFFORDABLES','0.25'),
(3,'CAMERA MAX','0.19');

INSERT INTO CAMERAS VALUES
(1,'CM51',3,18,25,1000),
(2,'A234',1,24,40,2000),
(3,'Z1',1,16,22,800),
(4,'J11',2,8,5,180),
(5,'A345',1,15,18,1250),
(6,'U25',2,14,21,800);

SELECT *FROM BRANDS;
SELECT *FROM CAMERAS;

SELECT NAME,CAMERAS.MODEL,CAMERAS.ZOOM,CAMERAS.PRICE
FROM BRANDS,CAMERAS
WHERE BRANDS.ID=CAMERAS.BRAND_ID;
```

```
SELECT BRANDS.NAME,CAMERAS.MODEL
FROM BRANDS JOIN CAMERAS ON BRANDS.ID=CAMERAS.BRAND_ID;

SELECT *FROM BRANDS JOIN CAMERAS ON BRANDS.ID=CAMERAS.BRAND_ID
WHERE MARKET_SHARE>0.2 AND ZOOM >=20;

SELECT NAME,MAX(CAMERAS.PRICE)
FROM BRANDS JOIN CAMERAS ON BRANDS.ID=CAMERAS.BRAND_ID
GROUP BY NAME;

SELECT NAME,CAMERAS.MODEL,(1-BRANDS.MARKET_SHARE)*CAMERAS.PRICE
FROM BRANDS JOIN CAMERAS ON BRANDS.ID=CAMERAS.BRAND_ID;
```

****

[Задача ООП]

Друга