

Софийски Университет “Св. Кл. Охридски”
Факултет по математика и информатика
Практически държавен изпит
Специалност Математика и информатика
21 март 2009

Студентите от специалност „Математика и информатика” решават задължително задача 4 и две задачи по избор измежду задачи 1, 2 или 3.

Решението на всяка задача се записва на отделен лист.

1. Елипсата $\varepsilon: \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$, където $a > b > 0$, има за фокуси точките F_1 и F_2 . Дадена е права l през точката F_1 , която не минава през F_2 и пресича ε в точките A и B . Допирателните към ε в точките A и B се пресичат в точка C .

а) Да се докаже, че CA и CB са ъглополовящите на външните ъгли на $\triangle ABF_2$;

б) Да се намери големината на $\angle CF_1B$.

2. За кои стойности на реалния параметър a системата

$$\begin{cases} x^2 - 4ax + 3 < 0 \\ -x^2 + 3x - 2 \geq 0 \end{cases}$$

има решение?

3. Върху страната AB на $\triangle ABC$ е взета точка D такава, че $BD = 4AD$. Центровете O_1 и O_2 на описаните съответно около $\triangle ADC$ и $\triangle BDC$ окръжности лежат на права p , успоредна на BC . Нека $T = p \cap DC$, $O = TB \cap CO_2$, а правата DO пресича страната BC в точка P . Да се намери лицето на $\triangle ABC$, ако лицето на $\triangle OBP$ е равно на 2.

4. Задача по информатика

Оценки:

2	от 0 до 54 точки
3	от 55 до 64 точки
4	от 65 до 74 точки
5	от 75 до 84 точки
6	от 85 до 100 точки

Напишете **синтактично правилно** следните класове на **Java** като **спазвате** концепциите за **скриване на информация** (*encapsulation, information hiding*) и **изискване за многократно използване на код** (*software reuse*-избягване на дублиране на код!) Типът на данните там, където не е изрично указан, да се избере съобразно естествените изисквания на задачата.

1. Напишете **class Time**, чиито обекти представляват **модел на** инстанция от време и имат данни **hh, mm и ss**, задаващи час, минути и секунди на времевата инстанция. Напишете пълен набор от конструктори (за общо ползване, по подразбиране и копиране) за **class Time** (по подразбиране час, минути и секунди са нула), **set** и **get** методи за данните на класа, а също **предефинирайте (override)** метода **toString()**на **class Object**. (20 точки)
2. Напишете **class Competition** , чиито обекти представляват **модел на** инстанция от състезание (*маратон, плувани и пр.*)- Всяка инстанция от състезание **има масив players с имената** на участниците, които се **дефинират при създаване** на инстанцията. Допълнително, всяко състезание **има и масив finishTimes** от обекти на **class Time** , съответстващи на показанията за време на елемент от масива с имена от участници. Нека **при създаване** на състезание **времето на всеки участник да се нулира**. Напишете конструктор за общо ползване за **class Competition** , **set** и **get** методи за данните **players** и **finishTimes** на класа, а също **предефинирайте (override)** метода **toString()**на **class Object**. (30 точки)
3. **Имплементирайте** стандартния **interface Comparable** в **class Time**, където методът

int compareTo(Object o)

да позволява наредба на инстанции от ***class Time*** в нарастващ ред
(10 точки)

4. Напишете в ***class Competition*** следния метод (12 точки)

public void finishCompetition(Time[] finish)

който инициализира времето на всеки от участниците в състезанието в съответствие с елементите на аргумента ***finish*** като преди това се прави проверка за валидност на масива ***finish***

5. Напишете в ***class Competition*** метод (14 точки)

public String getWinner()

който връща името на участника в състезанието с най- добро време съответстващо му в масива ***finishTimes*** като сравнението на времената се извършва посредством метода ***compareTo(Object o)*** на ***class Time***

6. Напишете ***class CompetitionTest***, който е конзолно приложение на Java за тестване на класовете ***Time- Competition***: (14 точки)

- a) създайте обект от ***class Competition*** с имена на трима участника,
- b) създайте масив ***testGame*** от три обекта ***Time*** с ненулеви час, минута и време
- c) изпълнете метода ***finishCompetition()*** като използвате масива ***testGame*** за аргумент
- d) изведете на стандартен изход името на участника в състезанието с най- добро време като използвате метода ***getWinner()*** на ***class Competition***

Забележка: Изпълнението на всеки клас да се пише на отделен лист като сорс кода се изписва четливо, ясно и в съответствие със синтаксиса на програмния език Java. Пълен брой точки се присъждат за пълно изпълнение на логиката на съответното задание и синтактично вярно решение.