

Objects in Java

1. Sumator

Направете клас с име Sumator. Той трябва да има методи за сумиране на две числа. Методите да имат едно и също име: sum(a, b). Входните параметри (a и b) да са от следните типове:

Integers;

Float;

Long

Short

Strings,

Varargs

java.math.BigInteger и

java.math.BigDecimal)

Метода трябва да принтира името си . В теста да се подават както примитиви пр. int, така и wrapper- Пр. Integer

2. Да се изгради и да се определят във възможно най-добра йерархия графичните фигури:

фигура, точка, окръжност, елипса и дъга

квадрат, правоъгълник, ромб, успоредник и многоъгълник

. Базов за всички фигури е абстрактния клас Figure. Да се измислят минимум 2 абстрактни метода и да се имплементират в наследниците. Методите трябва да разпечат низове при извикване – Пр. public void draw() {System.out.println("In Square")}

Да се имплементират default конструктори и конструктори с параметър определящ съответната геометрична фигура . Пр. Point(int x, int y)

Да се имплементира един init блок и един static блок в произволен клас (и да се разпечата при извикване). Да се проследи процеса на инициализация на обект от класа.

3. Магазин продава всякакви видове стоки. Да се изгради йерархия на продуктите, които се продават в магазина.

В магазинът има персонал (общ работник, продавач, чистач и т.н.). Има и клиенти. Да се построи йерархия на продуктите и персонала . Да се използват както интерфейси така и класове

а. Йерархията да има минимум 3 нива, общо класовете да са минимум 20, не повече от 30.

б. Всеки клас да има по минимум 2 метода и 2 атрибута .

в. Да се имплементират за всички класове методите toString(), equals и hashCode.

г.. Да се използва reflection в тестовият клас за да се извикат различните методи и да се

променят някои от private атрибутите.

4. Да се дефинира двоично дърво (Tree). Всички елементи на дървото (TreeElement) са от един и същ тип. Атрибути на елементите на дървото са цяло число, текстов низ и булева константа. Да се дефинира:

Хомогенно подредено дърво с $n > 11$ брой елементи

Операциите, които могат да се извършват върху дървото са:

добавяне на ново число към дървото. Ако има равни числа, последното да не се добавя

разпечатване на всички елементи на дървото сортирани

търсене на елемент в дървото

Да се реализира чрез използване на Comparator или Comparable

Атрибутите на елементите трябва да отговарят на Java Beans спецификацията (getters and setters) и да са енкапсулирани.

5. ЕГН Валидатор

Да се създаде програма ЕГН Валидатор – алгоритъм - <http://www.grao.bg/esgraon.html> .

При въвеждане на ЕГН програмата проверява дали е валидно и ако е отпечатва следната информация:

Дата на раждане:

Пол

Място на раждане – използвайте долното разпределение”

Благоевград от 000 до 043

Бургас от 044 до 093

Варна от 094 до 139

Велико Търново от 140 до 169

Видин от 170 до 183

Враца от 184 до 217

Габровоот 218 до 233

Кърджали от 234 до 281

Кюстендил от 282 до 301

Ловеч от 302 до 319

Монтана от 320 до 341

Пазарджик от 342 до 377

Перник от 378 до 395

Плевен от 396 до 435
Пловдив от 436 до 501
Разград от 502 до 527
Русе от 528 до 555
Силистра от 556 до 575
Сливен от 576 до 601
Смолян от 602 до 623
София - град от 624 до 721
София - окръг от 722 до 751
Стара Загора от 752 до 789
Добрич от 790 до 821
Търговище от 822 до 843
Хасково от 844 до 871
Шумен от 872 до 903
Ямбол от 904 до 925
Друг/Неизвестен от 926 до 999

След това можем да въведем число в дни - n.

Програмата изчислява на колко години, месеци и дни ще е човека със съответното ЕГН след n дни.

Програмата трябва да работи за хора родени преди 1900 и след 2000 г.

Да се използва DateFormat и Calendar класа.

Задача 6

Да се направи анотация @Copyright. Анотацията да може да се слага както на клас така и на метод. Тя има параметър type който е enum, съдържащ следните стойности: GPL, BSD, CC, COMMERCIAL.

Да се аностира клас CopyrightExample който има поне 5 празни метода.

Да се създаде програма CopyrightRunner, която взема аотируаният клас и проверява ако класа има анотация, но някои от методите няма то се изпринтва :Copyright class

<name> : (стойност от enum-а пр. GPL)

Ако метода има анотация то се принтира Copyright Method <name>(стойноста на конкретната анотация).

Ако класа няма анотация, но някои от методите има, то се изпринтва само съобщението за метода.

Пр.

```
@Copyright(type=CopyrightEnum.GPL)
```

```
class CopyrightExample {
```

```
void a() {
```

```
}
```

```
@Copyright(type=CopyrightEnum.BSD)
```

```
int b() {
```

```
}}
```

Изход:

Copyright Class **CopyrightExample: GPL**

Copyright Method **a: GPL**

Copyright Method **b: BSD**