**Klasy**  
    
  
**Zadanie 1 (4p)**

**Zaprojektować klasę**ZProstokat**reprezentującą prostokąty znakowe na konsoli. Dla każdego prostokąta należy określić: szerokość, wysokość, znak potrzebny do rysowania brzegu prostokąta na konsoli, znak potrzebny do rysowania wewnętrznej części prostokąta na konsoli. Szerokość i wysokość prostokąta powinny być liczbami naturalnymi dodatnimi (tj. od 1), natomiast w przypadku gdzy szerokość (lub wysokość) = 1 dwa wymienione znaki powinny być takie same.**

**Każdy prostokąt (lub w szczególnym przypadku, kwadrat) powinien mieć swój unikalny numer (zaczynając od 1), nadany automatycznie przy jego tworzeniu (**Pomoc: Skorzystać ze składowych statycznych klasy**). Prostokąty i kwadraty są numerowane oddzielnie.**

**Klasa**ZProstokat**powinna mieć metodę**public void rysuj()**drukującą informacje o prostokącie: typ (prostokąt/kwadrat), numer (prostokąta lub kwadratu), szerokość, wysokość, ilość znaków wewnętrznych, ilość znaków brzegowych oraz "konsolową" postać znakową prostokąta.**

**W klasie**ZProstokat**należy skorzystać z wywołania kontruktora 4-parametrowego z innego konstruktora, 3-parametrowego za pomocą słowa kluczowego***this***.**

**Stworzyć w/w klasę w taki sposób, aby następujący program:**

public class ZProstokatTest {  
  
    public static void main(String[] args) {  
          
            ZProstokat zp[] = {  
                new ZProstokat(4, 'a', 'e'),        // konstruktor z 3 parametrami  
                new ZProstokat(5, 3 , '\*', '+'),   // konstruktor z 4 parametrami      
                new ZProstokat(1, 2, 'a','a'),  
                new ZProstokat(3, 3, '+', 'x'),  
                new ZProstokat(1, 2, 'x', 'y'),  
                new ZProstokat(3, 4, '^', '$')  
            };  
              
            for (ZProstokat z : zp)  
                try {  
                    z.rysuj();  
                } catch(Exception e) {  
                    System.out.println(e.getMessage());  
                }  
    }  
}

**wyprowadził na konsolę poniższe wyniki:**

Kwadrat (1) rozmiaru 4 x 4,  a = 4, e = 12  
e e e e   
e a a e   
e a a e   
e e e e   
  
Prostokąt (1) rozmiaru 5 x 3,  \* = 3, + = 12  
+ + + + +   
+ \* \* \* +   
+ + + + +   
  
Prostokąt (2) rozmiaru 1 x 2,  a = 2  
a   
a   
  
Kwadrat (2) rozmiaru 3 x 3,  + = 1, x = 8  
x x x   
x + x   
x x x   
  
Prostokąt (3) rozmiaru 1 x 2  
Błędny prostokąt!  
  
Prostokąt (4) rozmiaru 3 x 4,  ^ = 2, $ = 10  
$ $ $   
$ ^ $   
$ ^ $   
$ $ $  **ZADANIE 2 (6p)**  **Zaprojektować klasy:**Kurs**reprezentującą kursy oraz**Osoba**reprezentującą uczestników kursów. Każda osoba-uczestnik posiada imię, wiek. Każdy kurs posiada nazwę, dolny limit wiekowy uczestników, górny limit liczby uczestników oraz strukturę (np. tablicę) przechowującą uczestników.  
Na kurs można zapisywać osobę (stanie się ona uczestnikiem kursu) według następujących kryteriów (kolejność od najważniejszego): czy podana osoba już jest uczestnikiem kursu, czy podany kurs ma jeszcze wolne miejsce, czy osoba spełnia kryterium wiekowe kursu. Z kursu można również usunąć uczestnika.  
  
Podanie**null**jako argumentu dla osoby/uczestnika w wywołaniu metod powinno spowodować obsługę wyjątku.**  
  
**Klasa**Kurs**powinna mieć metodę**public String toString() **zwracajacą informacje o kursie: nazwę, uczestników.**  
**Stworzyć w/w klasy w taki sposób, aby następujący program:**  
  
public class KursTest {  
  
    public static void main(String[] args) {  
                 
            // utworzenie kursu z nazwą, dolnym limitem wiekowym (np. 3), górnym limitem uczestników (np. 4)   
        Kurs java = new Kurs("Java", 3, 4), cpp = new Kurs("C++", 2, 2);     
             
            // tablica kursów  
        Kurs[] kursy = {java, cpp};  
         
            // utworzenie potencjalnych uczestników kursów z imionem i wiekiem  
        Osoba janek = new Osoba("Janek", 6), adas = new Osoba("Adaś", 3), aleks = new Osoba("Aleks", 2), zosia = new Osoba("Zosia", 4), lenka = new Osoba("Lenka", 5);  
      
        try {  
            java.dodajUczestnika(adas);    // próba zapisywania osoby (np. adas) na kurs (np. java)  
            java.dodajUczestnika(janek);  
            java.dodajUczestnika(aleks);  
            java.dodajUczestnika(zosia);  
            java.dodajUczestnika(lenka);  
            java.dodajUczestnika(lenka).dodajUczestnika(aleks);  
         
            System.out.println();  
             
            cpp.dodajUczestnika(aleks);    // próba zapisywania osoby (np. aleks) na inny kurs (np. cpp)  
            cpp.dodajUczestnika(janek);  
             
            System.out.println();  
             
            java.dodajUczestnika(null);    // błędnie podana osoba przy zapisywaniu na kurs  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
             
        System.out.println("\n" + java + "\n");  
      
        try {  
            java.usunUczestnika(aleks);        // próba usunięcia uczestnika z kursu  
            java.usunUczestnika(janek);  
      
            System.out.println();  
      
            java.usunUczestnika(null);         // błędnie podana osoba przy usunięciu  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
         
        System.out.println("\n" + java);  
        System.out.println(cpp + "\n");  
         
            // najliczniejszy(...) jest metodą statyczną klasy KursTest, zwracającą kurs z największą liczbą uczestników  
            // trzeba dopisać tę metodę  
        System.out.println("Kurs z największą liczbą uczestników: " + najliczniejszy(kursy).getNazwaKursu());  
    }  
  
}  
  
**wyprowadził na konsolę poniższe wyniki:**  
Dodano uczestnika Adaś do kursu Java  
Dodano uczestnika Janek do kursu Java  
Aleks nie spełnia kryterium wiekowego do kursu Java  
Dodano uczestnika Zosia do kursu Java  
Dodano uczestnika Lenka do kursu Java  
Lenka już jest zapisany do kursu Java  
Kurs Java: brak miejsc!  
  
Dodano uczestnika Aleks do kursu C++  
Dodano uczestnika Janek do kursu C++  
  
Nie podano uczestnika przy zapisywaniu!  
  
Kurs Java, uczestnicy: Adaś Janek Zosia Lenka   
  
Kurs Java nie ma uczestnika Aleks  
Usunięto uczestnika Janek z kursu Java  
  
Nie podano uczestnika przy usunięciu!  
  
Kurs Java, uczestnicy: Adaś Zosia Lenka   
Kurs C++, uczestnicy: Aleks Janek   
  
Kurs z największą liczbą uczestników: Java