

I. Да се състави клас Color, имплементиращ интерфейс Comparable:

Скрити член променливи

съставни елементи по основните цветове (RGB) - long и

общ (смесен) цвят, образуван от основните по формула:

$256 * RValue + 256 * GValue + 1 * BValue$

Публични членове:

Конструктори:

1) Подразбиращ се: `public Color() {}`

2) Експлицитен-по общ цвят: `public Color(long c)`

Методи:

4) четене/запис - за съставните цветове по 3 get/set

5) Стрингова интерпретация на обектите -toString()

6) Еквивалентност с подадения параметър-equals(Object r)

7) Сравнение "<" compareTo(Object c) по избор

II. Да се състави клас ColorRectangle в двумерното пространство, наследник на клас Color,

имплементиращ интерфейс Comparable:

Скрити член променливи, координати - int iX1, iY1, iX2, iY2;

Публични членове:

Конструктори:

1) Подразбиращ се:

2) Експлицитен - координати и съставен цвят;

Методи:

4) четене/запис по 4 get/set

5) Изчислява/върща площ на правоъгълника int calcArea()

6) Сравнява с подадения като параметър за "по-малък" по площ -1,0,+1
int compareTo(ColorRectangle r) или (public int compareTo(Object r))

7) Стрингова интерпретация на обектите-String toString()

8) Еквивалентност с подадения параметър-boolean equals(...),

включително цвят

9) Изместване по X на подадените точки:

`public void translateX(int iPoints)`

10) Изместване по Y на подадените точки

`public void translateY(int iPoints)`

10) Изместване по X и Y на подадените точки

11) Точки X и Y в правоъгълника ли са- boolean isInside(int ptX, int ptY)

12) Изчислява/върща обграждащ правоъгълник между подадения параметър и члена

13) Изчислява/върща общ правоъгълник между подадения параметър и члена

III. Главна функция:

Създава обекти, извежда:

Измества по X и Y, извежда,

Проверява за еквивалентност, извежда

Проверява за по-малко, извежда

Изчислява обграждащ правоъгълник, извежда

Изчислява общ правоъгълник, извежда

Задача за упражнение 4, при доц. Николов

I. Да се състави клас: MainClass с главна функция, която:

I.1. Да създава масиви от прости типове данни:

boolean[];

byte[];

char[];

short[];

int[];

long[];

float[];

double[];

String[];

-Да се запълнят със еднакви данни (Arrays.fill)

-Да се инициализират с различни данни;

-Да се сортират (Arrays.sort);

-Да се изведе информацията на конзолата (Arrays.toString);

I.2. Да се съставят масиви от класовете Color и ColorRectangle;

-Да се запълнят с примерни данни

-Да се сортират (Arrays.sort)

-Да се изведе информацията на конзолата (Arrays.toString):

II. Да се състави клас RectangleArray, съдържащ частен масив от обекти ColorRectangle

II.1. Конструктор - експлицитен с име на файл: RectangleArray(String fileName)

10 правоъгълника. Формат:

<x1>SP<y1>SP<x2>SP<y2><SP><color>CR

Примерно съдържание на файла:

1 1 2 3 255

1 2 3 3 255

1 3 4 5 255

1 4 5 6 255

1 5 2 3 255

1 6 3 3 255

1 7 4 5 255

1 8 5 6 255

1 9 2 3 255

1 10 3 3 255

Публични методи:

II.2. Извеждане на конзолен изход - void printColl()

II.3. Изчисляване на сумарна площ - double calcSumArea()

II.4. Изчисляване на сумарен периметър - double calcSumPerimeter()

II.5. Тестова главна функция, която:

- да създава обект от файл, съдържащ 10 реда с примерни данни за правоъгълници, извежда;

-да сортира масива, извежда;

-да изчисли сумарна площ, извежда;

-да изчисли сумарен периметър, извежда.

Задача за упражнение 09. Тема – колекции, алгоритми върху колекции

Да се състави клас съхраняващ множество от правоъгълници: *RectangleCollection*:

за съхраняване на *ColorRectangle*:

Член (частен) интерфейс *SortedSet*, *map TreeSet* за съхраняване на множество правоъгълници

* Конструктори/Методи:

* 1.1) Подразбиращ се

* 1.2) Експлицитен по име на файл: *RectangleCollection(String)*

* Файлът съдържа правоъгълници. Формат: *x1 y1 x2 y2 color*

* Example:

-10 -10 6 10 255

-1 -1 10 6 255

-2 -2 10 6 255

-3 -1 10 6 255

-1 -1 10 6 255

* 1.3) Добавя подадения параметър към колекцията *void addRectangle(...)*

* 1.4) Изчислява заграждащ правоъгълник: *ColorRectangle calcUnionColl()*

* 1.5) Изчислява минимален по площ правоъгълник: *ColorRectangle calcMin()*

* 1.6) Изчислява максимален по площ правоъгълник: *ColorRectangle calcMax()*

* 1.7) Изчислява/върща броя на припокритите от някой друг правоъгълник правоъгълници *int calcNested()*

* 1.8) Върща броя елементи в множеството *int calcSize()*

* 1.9) Извежда колекцията *void printColl()*

* 1.10) Изчислява/върща суматна площ *double calcSumArea()*

* 1.11) Изчислява/върща сумарен периметър *double calcSumPerimeter()*

* 1.12) Изчислява/върща списък с обратен ред *List reverseList()*

* 1.13) Търси правоъгълника параметър *boolean findRect(ColorRectangle toFind)*

Главна функция

* Създава обект *RectangleCollection* по зададен входен файл *rects.txt*, добавя елемент 2, 2, 400, 400, 255, извежда

* Изчислява обграждащ правоъгълник

* Изчислява мин. правоъгълник

* Изчислява макс. правоъгълник

* Изчислява броя на припокритите правоъгълници

* Изчислява списък в обратен ред

* Изчислява сума от площи

* Изчислява сума от периметри

* Изчислява размерност

Задача за упражнение 10 Тема - колекции, интерфейси за файлове вход/изход, алгоритми

I. Да се състави интерфейс *IFile* методи:

boolean remove() void save()

void load()

II. Да се промени клас *RectangleCollection* съхраняващ множество от правоъгълници (упр. 9), който да имплементира интерфейс *IFile*:

II.1. Добавяне на 2 полета (частни) за съхраняване на имена на входен и изходен файл.

Конструктори/методи:

II. 2) Експлицитен конструктор по 2 имена на файлове – за вход и за изход

Файловете да осигуряват съхраняване на съдържанието на правоъгълниците в еднакъв формат:

<x1>SP<y1>SP<x2>SP<y2><SP><color>CR

Пример:

-10 -10 6 10 255