Упражнение 2. Създаване и тестване на класове. Статични методи

Цел на упражнението

Това упражнение е посветено на създаване на класове и тяхното тестване посредством JUnit тестващо приложение, както и на работа със статични методи.

Задача 1. Свалете си файла ThreeDimPoint.java от сайта на обучението.

Съхранете го в нова директория.

Задача 2. Разгледайте началото на класа ThreeDimPoint.

Всяка инстанция на класа ThreeDimPoint е тридименсионна като представя точката (x, y, z). Класът има 3 полета x, y и z, конструктор и три getter функции. Разучете какво правят тези четири метода ОТ ТЕХНИТЕ СПЕЦИФИКАЦИИ.

Задача 3. Създайте тестов клас и изтествайте конструктора и трите gette функции.

В методите има грешки. НЕ СЕ ОПИТВАЙТЕ ДА ГИ ТЪРСИТЕ. Целта на упражнението е откриете грешките посрдством тест. Затова създайте JUnit test class и изтествайте 4 метода. Продължавайте теста докато ме получите съобщение за грешка. Ето какво трябва да направите:

1. В DrJava, изберете меню File елемент New JUnit Test Case.... Това действие ще създаде втори файл, дайте му име DimTester.java. Съхранете го в същата директория, където е и ThreeDimPoint.java.

2. Променете името на метода testX в класа DimTester на testConstructor. Променете и коментарите в javadoc подходящо.

3. В тялото на метода testConstructor напишете Java код, който:

   (a) Създава инстанция на ThreeDimPoint и записва името й в променлива. Забележете, че извикването на конструктора изисква три цели аргумента - x, y и z-координати на точката.

   (b) Вмъкнете тестове, за да проверите, че конструктора задава правилна стойност на полетата. За целта използвайте метода assertEquals. Спомнете си, че наследената процедура assertEquals(expected,computed) спира всички тестове и изходи ако expected!=computed.

4. Компилирайте двата класа, чрез бутон Compile.

5. Натиснете бутон Test.

6. Ако се получи грешка, разгледайте съобщението и конструктора, както и функцията getter, която се използва, за да определите къдее грешката. Поправете я.

Повторете стъпки 4, 5 и 6 докато не изчистите програмата от грешки.

Задача 4. Тестова функция hasAZero.

Според спецификацията, тази функция трябва да върне true, ако поне една от координатите x, y, или z е 0. Ако нито една не е нула, функцията връща false. Не правете промени по функцията преди тестване. В класа DimTester, напишете друга тест функция, testHasAZero, която ще тества функцията hasAZero. Един възможен тест е всички възможни комбинации на множеството от стойности (x, y, z). Помислете, колко тестови ситуации трябва да направите, за да сте сигурни, че метода е правилен. Реализирайте ги в тестовата процедура testHasAZero. Тя трябва да създаде повече от една инстанция на ThreeDimPoint, за да ги реализира.

Компилирайте тестовата програма и я стартирайте. Ако има грешки, ги поправете. Продължете тестването, докато не получите програма без грешки.

Задача 5. Направете статичен метод.

Упражнение 2. Създаване и тестване на класове. Статични методи

Дължината на линия от точка (0, 0, 0) до точка (x, y, z) се изчислява по $\sqrt[2]{x^2 + y^2 + z^2}$. В Java, има функция Math.sqrt(n), която намира $\sqrt[2]{n}$, както е показано в метода length на класа ThreeDimPoint. Той изчислява дължината на линия от началото до точката, записана в инстанцията. Извикайте я в Interactions pane чрез:

ThreeDimPoint d;

d= new ThreeDimPoint(3, 4, 5);

d.length();

Сега направете същото в един ред:

(new ThreeDimPoint(3,4,5)).length();

Вторият начин показва израз new, който НЕ се присвоява на променлива. Това е допустимо, стига да не се нуждаем от по-нататъшни обръщения към тази инстанция. Създава се нова папка (обект) и се записва във файла-чекмедже ThreeDimPoint след което се извиква функцията length() на този обект.

Направете функцията статична като запишете думата static след думата public и компилирайте програмата. Какво се случва? Какъв е проблема? След това премахнете думата static и компилирайте отново. Има ли промяна?

Задача 6. Направете метода static.

Нека разгледа заглавието на функция length1:

public double length1(ThreeDimPoint c)

Когато тя се извиква, има един аргумент от тип ThreeDimPoint. Направете метода статичен, чрез добавяне на думата static след public и компилирайте. Програмата се компилира без проблем. Защо?

Защото методът не се отнася директно до кое и да е поле на инстанцията, в която е дефиниран метода. Следователно, той не се нуждае да бъде във всяка инстанция и може да бъде статичен. Има САМО ЕДНО негово копие и то е в ThreeDimPoint.

Изпълнете редовете, показани по-долу, в Interactions pane, един по един. Първият създава папка ThreeDimPoint и записва нейното име в променливата f. Вторият връща дължината на линията, чрез извикване на функцията length на обекта f. Третият връща дължината на линията, чрез извикване на статичната функция ThreeDimPoint.length1.

ThreeDimPoint f= new ThreeDimPoint(3,4,5);

f.length()

ThreeDimPoint.length1(f)

Тук се илюстрира как се използва статичен метод. Ако един метод не реферира полета на обекта, неговот място е във файла-чекмедже като го направим static.

Задача 7. Класът Math.

Класът Math има много статични функции, които изпълняват различни математически операции. Той е достъпен винаги без използване на import оператор. Съхранява статични променливи и статични функции. Няма полета, които могат да се сменят. Няма методи на инстанциите. Всичко при него е статично. Ето някои функции и тяхното предназначение:

Math.abs(x)	Абсолютна стойност на x	Math.abs(-4) е 4
Math.sqrt(x)	Корен квадратен на x	Math.sqrt(25) е 5.0
Math.min(b,c)	По-малкото от b и c	Math.min(5, 4) е 4
Math.max(b,c)	По-голямото от b и c	Math.max(–6,4) е 4

Упражнение 2. Създаване и тестване на класове. Статични методи

Math.floor(x)	Най-голямото цяло, което не е по-голямо от x	Math.floor(–3.2) е –4.0 Math.floor(3.2) е 3.0
Math.ceil(x)	Най-малкото цяло, което не е по-малко от x	Math.ceil(–3.2) е –3.0 Math.ceil(3.2) е 4.0
Math.PI	best double approximation to pi, the ratio of the diameter of a circle to its circumference	

Опитайте как работят в Interactions pane следните функции:

Math.min(-7, 4)

Math.min(Math.min(3,4), 5)

Math.sqrt(5)

Math.sqrt(-5)

Math.floor(-3.7)

Math.ceil(3.7)

Math.ceil(-3.7)

Math.abs(3.7)

Math.abs(-3.7)

Math.abs(-3)

Math.PI

Задачи за самостоятелна работа

1) Напишете функция, която връща true, ако първият или последният елемент на масива е 6
```
public boolean firstLast6(int[] nums) {

}
```
2) Напишете функция, която по зададен низ връща нов, в който всеки символ е дублиран
```
public String doubleChar(String str) {


}
```
3) Ще наречем едно число специално, ако е кратно на 11 или е с едно по-голямо (напр. 22 или 23). Напишете функция, която по зададен параметър връща true, ако е специално число.
```
public boolean specialEleven(int n) {

}
```
4) Ще казваме, че един низ е xy балансиран, ако за всяко x има поне едно y след него независимо на коя позиция в низа. Няколко x могат да се балансират с един y. Напишете функция, която по зададен низ параметър, връща дали е xy балансиран.
```
public boolean xyBalance(String str) {

}
```
5) Напишете функция, която преброява xx в низ параметър. Допуска се припокриване, т.е. xxx дава 2.
```
int countXX(String str) {

}
```