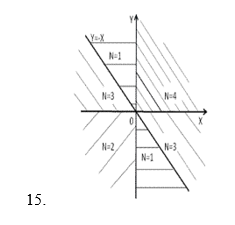
ЛАБОРАТОРНА РОБОТА №1

Написати програму, яка визначає номер області, до якої належить точка M з заданими координатами. Межі областей відносити до області з найбільшим номером. Варіант завдання: 15.



Розв'язання:  
  
public class IndividualTask {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.in);  
 int N = 0;  
 double x, y;  
  
 System.out.print("Введіть координату X для точки M: ");  
 if (input.hasNextDouble()) {  
 x = input.nextDouble();  
 } else {  
 System.out.println("Помилка: введено нечислове значення для X.");  
 return;  
 }  
  
 System.out.print("Введіть координату Y для точки M: ");  
 if (input.hasNextDouble()) {  
 y = input.nextDouble();  
 } else {  
 System.out.println("Помилка: введено нечислове значення для Y.");  
 return;  
 }  
  
 if (y >= 0) {  
 if (y >= x) {  
 N = 1;  
 } else {  
 N = 4;  
 }  
 } else {  
 if (y >= x) {  
 N = 2;  
 } else {  
 N = 3;  
 }  
 }  
  
 System.out.println("Результат:");  
 System.out.println("Точка M(" + x + ", " + y + ") належить області N = " + N);  
 }  
}

**1.Яке призначення директорій src/main/java та src/main/resources у Javaпроєкті?**

**src/main/java** — каталог для основного вихідного коду Java (класи, інтерфейси, тощо).

**src/main/resources** — каталог для некомпільованих ресурсів, таких як конфігураційні файли, шаблони, зображення, які потім потрапляють до фінального .jar чи .war файлу.

**2. Яку роль відіграє файл pom.xml у проєктах Maven?**

*Файл* ***pom.xml*** *описує конфігурацію Maven-проєкту: його залежності, плагіни, цілі компіляції, структуру збірки, тощо. Він є основним центром управління проєктом.*

**3. Яку роль відіграє файл build.gradle у проєктах Gradle?**

*Файл* ***build.gradle*** *визначає, як Gradle має зібрати проєкт: залежності, завдання (tasks), конфігурації плагінів. Це аналог pom.xml для Gradle, але має більш гнучкий і програмований підхід.*

**4. Що таке залежність у контексті Java-проєкту? Наведіть приклад.**

***Залежність*** *— це бібліотека або модуль, який потрібен вашому проєкту для роботи.*

** ***Приклад****: якщо ви хочете працювати з JSON у Java, можна додати залежність на бібліотеку Gson:*

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.8.9</version>

</dependency>

**5. Які переваги використання систем управління залежностями, таких як Maven або Gradle?**

*-Автоматичне завантаження та оновлення бібліотек.*

*-Просте управління версіями залежностей.*

*-Стандартизована структура проєкту.*

*-Полегшення процесу збірки, тестування та розгортання.*

*-Велика кількість плагінів і розширень для автоматизації завдань.*

**6. Опишіть покроковий процес додавання нової залежності до Maven проєкту в IntelliJ IDEA.**

*- Відкрити проєкт у IntelliJ IDEA.*

*-Відкрити файл pom.xml.*

*-Додати новий <dependency> у секцію <dependencies>.*

*-Натиснути* ***"Load Maven Changes"*** *або натиснути спливаюче повідомлення для оновлення проєкту.*

*-Maven завантажить потрібні бібліотеки автоматично.*

**7. Чим відрізняється додавання залежності в Maven від додавання залежності в Gradle?**

*У* ***Maven*** *залежності додаються у форматі XML у pom.xml.*

*У* ***Gradle*** *залежності додаються в секцію dependencies у build.gradle у стилі коду (Groovy або Kotlin DSL),*

*Gradle має більш декларативний і скриптовий стиль, а Maven — суворо структурований XML-стиль.*