**🧠RooCodeによるJavaコード生成チュートリアル**

**🔧 事前準備：プロジェクトの作成とファイル準備**

1. **VSCodeを起動**
2. 左のメニューから「エクスプローラー」アイコンをクリック
3. 適当なフォルダを開く（例：MyJavaProject）
4. srcフォルダを作成し、右クリック → 「新しいファイル」→ RectangleUtils.java と入力して作成
5. 以下のようなテンプレートを入力しておく（RooCodeはこのような文脈があると精度が上がります）：

public class RectangleUtils {  
   
 // ここに面積を計算するメソッドを生成したい  
   
}

**🚀 コード生成のやり方（パターン別）**

**🅰️ パターン1：プロンプトで新規メソッド生成**

1. 「// ここに〜」の行にカーソルを合わせる
2. Ctrl + Shift + P を押して「コマンドパレット」を開く
3. RooCode: Generate Code を選択（🔍で検索してEnter）
4. 以下のように自然言語でプロンプトを入力：

幅と高さを引数に取り、面積（int型）を返すメソッドをJavaで書いて。

1. 数秒後、以下のようなコードが挿入されます：

public static int calculateArea(int width, int height) {  
 return width \* height;  
}

**🅱️ パターン2：コメントから文脈推論（事前プロンプトなし）**

1. コメントとして日本語で意図を書く：

// 幅と高さから面積を返す処理が欲しい

1. そのコメント行にカーソルを合わせ、同様に RooCode: Generate Code を実行
2. コメントの意味を読み取ってコードが自動で生成される（プロンプト入力不要）

📌 **この方法は既存コードに追記する場合や、手が空いた部分にヒントだけ書いておくと便利です。**

**🆎 パターン3：クラスごと自動生成**

1. 新しいファイル（例：User.java）を作成
2. 何も入力せず、ファイルの先頭で RooCode: Generate Code を実行
3. プロンプトにこう入力：

ユーザーのID（int）、名前（String）、メールアドレス（String）を持つJavaのクラスを作って。

1. 以下のようなクラスが生成される：

public class User {  
 private int id;  
 private String name;  
 private String email;  
  
 public User(int id, String name, String email) {  
 this.id = id;  
 this.name = name;  
 this.email = email;  
 }  
  
 // getter, setterもつくことがある（明示すればより確実）  
}

**🅾️ パターン4：既存コードの補完・修正**

1. 以下のように中途半端なメソッドを書いておく：

public static int calculateArea(int width, int height) {  
 // 面積を返す  
}

1. コメントに合わせて RooCode: Generate Code を実行
2. 「面積を計算して返す処理を補って」と入力すると、不足分が補完されます：

return width \* height;

**🔄 パターン5：複数メソッドの生成（プロンプト工夫）**

1. 例えばこのようなプロンプト：

Javaで長方形の周囲長を返すメソッドと、面積を返すメソッドを両方書いて。

1. 結果として、複数のメソッドが一度に出力されることがあります：

public static int calculateArea(int width, int height) {  
 return width \* height;  
}  
  
public static int calculatePerimeter(int width, int height) {  
 return 2 \* (width + height);  
}

**📌 補足：プロンプトの書き方のコツ**

|  |  |
| --- | --- |
| よくある表現 | 意味すること |
| 「〜を引数に取って」 | パラメータ指定（型と名前を書くと精度UP） |
| 「〜を返す」 | 戻り値の型を明示 |
| 「クラスを作って」 | POJOの生成 |
| 「getter/setterつけて」 | LombokなしのJavaBean風 |
| 「例外処理も書いて」 | try-catch構造の挿入 |

**✅ 成功確認**

すべて完了したら、以下のように実行テストをしてみましょう：

public class Main {  
 public static void main(String[] args) {  
 System.out.println(RectangleUtils.calculateArea(5, 7)); // → 35  
 }  
}