**Exercise 2: Implementing the Factory Method Pattern**

1. **Document Interface (Abstract Product)**:
   * Defines the common operations all documents must implement (open, save, close)
2. **Concrete Document Classes**:
   * WordDocument, PdfDocument, and ExcelDocument implement the Document interface
   * Each provides specific implementations of the document operations
   * They can also include document-specific methods (edit(), preview(), calculate())
3. **DocumentFactory Abstract Class (Creator)**:
   * Declares the factory method createDocument()
   * May contain default implementations of operations that work with products
   * The processDocument() method demonstrates how the factory can work with the product
4. **Concrete Factory Classes**:
   * Each factory (WordDocumentFactory, PdfDocumentFactory, ExcelDocumentFactory) implements the createDocument() method
   * Each factory is responsible for creating a specific type of document
5. **Main Class (Client Code)**:
   * Demonstrates how to use the factories to create documents
   * Shows both direct creation and processing through factory methods
   * Demonstrates how to access document-specific methods through casting

Benefits of This Implementation

1. **Flexibility**: New document types can be added without modifying existing client code
2. **Encapsulation**: The creation logic is encapsulated in factory classes
3. **Single Responsibility**: Each factory is responsible for creating only one type of document
4. **Open/Closed Principle**: The system is open for extension (new document types) but closed for modification

**Output**

