

Abstracte Klassen und Vererbung:

1. **Erstellen einer abstrakten Klasse Shape:**
 - Erstelle eine abstrakte Klasse namens Shape, die als Basis für geometrische Formen dient.
 - Implementiere eine abstrakte Methode namens calculateArea() für die Berechnung der Fläche.
2. **Ableitung von konkreten Formen:**
 - Leite die Klassen Circle und Rectangle von der abstrakten Klasse Shape ab.
 - Implementiere die calculateArea()-Methode für jede Form, um die Fläche entsprechend zu berechnen.

Interfaces und Vererbung:

3. **Entwurf eines Fahrzeug-Interfaces:**
 - Definiere ein Interface namens Vehicle, das grundlegende Methoden für Fahrzeuge bereitstellt, wie start(), stop() und accelerate().
 - Implementiere Klassen wie Car, Bike, und Truck, die das Interface Vehicle umsetzen und individuelle Funktionalitäten für jedes Fahrzeug bieten.
4. **Erstellung eines GeometricObject-Interfaces:**
 - Entwickle ein Interface namens GeometricObject, das Methoden wie calculateArea() und calculatePerimeter() definiert.
 - Implementiere die Klassen Circle und Square, um dieses Interface zu realisieren und die entsprechenden Methoden zu implementieren.

Modifier:

5. **Kontrollieren des Zugriffs auf Attribute:**
 - Erstelle eine Klasse Person mit privaten Attributen wie name, age und address.
 - Verwende public und private Modifier, um den Zugriff auf diese Attribute zu steuern.
 - Implementiere öffentliche Methoden, um auf die Attribute zuzugreifen und sie zu aktualisieren.
6. **Sicheres Attributzugriff mit protected:**
 - Definiere eine Klasse BankAccount mit einem Attribut balance.
 - Verwende den protected Modifier für das balance-Attribut, um den Zugriff auf die Klasse selbst und abgeleitete Klassen zu beschränken.

Kombinierte Konzepte:

7. **Musikinstrumente spielbar machen:**
 - Erstelle ein Interface Playable mit einer Methode play().
 - Implementiere eine abstrakte Klasse Instrument, die das Interface implementiert.
 - Leite Klassen wie Guitar und Piano von Instrument ab und implementiere die play()-Methode für jedes Instrument.
8. **Entwicklung eines Online-Shopping-Systems:**
 - Nutze abstrakte Klassen oder Interfaces wie Product, User und Cart.
 - Implementiere Methoden, die es einem Benutzer ermöglichen, Produkte zum Warenkorb hinzuzufügen, zu entfernen usw.

Fortgeschrittene Aufgaben:

9. Charaktere in einem Videospiel:

- Entwickle ein System für ein Videospiel mit verschiedenen Charakteren.
- Verwende abstrakte Klassen oder Interfaces, um Gemeinsamkeiten zwischen den Charakteren zu definieren.
- Erstelle konkrete Klassen für verschiedene Arten von Charakteren und ihre spezifischen Eigenschaften.

10. Verwaltung von Fahrzeugen:

- Erstelle eine Anwendung zur Verwaltung verschiedener Fahrzeuge.
- Nutze Vererbung, abstrakte Klassen und Interfaces, um Gemeinsamkeiten und Unterschiede zwischen den Fahrzeugen zu modellieren und zu verwalten.