

1. **继承**：它允许一个类（子类）继承另一个类（父类）的属性和方法。

- 作用：

- 代码复用：子类可以直接使用父类中已经实现的方法和属性，避免重复编写代码。

- 层次结构构建：构建出类的层次结构，便于对不同类型的对象进行统一管理。

2. **接口**：定义了一组方法的规范，类可以实现一个或多个接口来实现这些方法。

- 作用：

- 定义行为规范：明确了实现类应该具备的方法。

- 实现多态：不同的类可以实现同一个接口，通过接口变量可以引用不同的实现类对象，实现多态。

3. **多态**：是指同一个行为具有多种不同的表现形式。在 Java 中，多态主要通过继承和接口实现。

- 作用：

- 增加代码的灵活性：可以根据不同的对象类型动态地调用相应的方法，提高了代码的可扩展性和可维护性。

- 提高代码的可复用性：可以使用父类或接口类型的变量来引用不同的子类对象，从而实现代码的复用。

4. **泛型**：泛型允许在定义类、接口和方法时使用类型参数，使得代码可以在不同的数据类型上进行复用，也提高了代码的安全性。

- 作用：

- 类型安全：在编译阶段就可以检查类型错误，避免了运行时的类型转换错误。

- 代码复用：可以编写通用的代码，适用于不同的数据类型。



