

Sealed classes

Sealed Classes

- Used to restrict which other classes may directly extend your class
- New keywords: `sealed`, `non-sealed`, `permits`
- Sealed classes must be declared in the same package as its direct subclasses
- Every listed subclass must extend the sealed class
- Every subclass must have one of these class modifiers:
 - `final`, `sealed`, `non-sealed`

```
public sealed class Car permits Ford, Renault, Fiat {}
```

```
// each of these classes must extend Car
```

```
// subclasses have to be made either final, sealed or non-sealed
```

```
public final class Ford extends Car {}
```

```
// cannot be further extended
```

```
public non-sealed class Renault extends Car {}
```

```
// can be extended by any class
```

```
public sealed class Fiat extends Car permits Uno, Punto {}
```

```
// can be extended, but only by classes Uno and Punto
```

// interfaces can also be made sealed:

```
public sealed interface Mammal permits Cat, Dog, Eats {}
```

// Cat, Dog and Eats could be classes or interfaces

```
public non-sealed interface Eats extends Mammal {}
```

// interfaces cannot be final!

// classes must implement interface Mammal directly or indirectly

```
public final class Cat implements Mammal {}
```

```
public final class Dog implements Eats {}
```

// implements Mammal through Eats

