What is the 'annotations' package?

This package implements compilation of Angular-annotated classes - those with `@Component`, `@NgModule`, etc. decorators. (Note that the compiler uses 'decorator' and 'annotation' interchangeably, despite them having slightly different semantics).

The 'transform' package of the compiler provides an abstraction for a `DecoratorHandler`, which defines how to compile a class decorated with a particular Angular decorator. This package implements a `DecoratorHandler` for each Angular type. The methods of these `DecoratorHandler`s then allow the rest of the compiler to process each decorated class through the phases of compilation.

Anatomy of `DecoratorHandler`s

Each handler implemented here performs some similar operations:

- It uses the `PartialEvaluator` to resolve expressions within the decorator metadata or other decorated fields that need to be understood statically.
- It extracts information from constructors of decorated classes which is required to generate dependency injection instructions.
- It reports errors when developers have misused or misconfigured the decorators.
- It populates registries that describe decorated classes to the rest of the compiler.
- It uses those same registries to understand decorated classes within the context of the compilation (for example, to understand which dependencies are used in a given template).
- It creates `SemanticSymbol`s which allow for accurate incremental compilation when reacting to input changes.
- It builds metadata objects for `@angular/compiler` which describe the decorated classes, which can then perform the actual code generation.

Since there is significant overlap between `DecoratorHandler` implementations, much of this functionality is implemented in a shared 'common' sub-package.