According to angular.io “Angular applications are modular and Angular has its own modularity system called NgModules.” (*Angular*, n.d.) The NgModules contain blocks of code that are part of the application domain or workflow. NgModules have components, service providers or other bits of code that is part of the NgModule. (*Angular*, n.d.) “They can import functionality that is exported from other NgModules” (*Angular*, n.d.) and they export other parts of the NgModules that are to be used by more NgModules. (*Angular*, n.d.) A smaller application may have just a single NgModule or a few, but most of the time applications will have multiple modules. (*Angular*, n.d.)

Each application for Angular has to have one NgModule class, also called the root module. The root module is named AppModule and is found in the file called app.module.ts (*Angular*, n.d.) The application is launched from bootstrapping the AppModule. (*Angular*, n.d.)

In order to define the NgModule class it needs to have this @NgModule(). (*Angular*, n.d.) This is called a decorator and it’s a function. “takes a single metadata object, whose properties describe the module.” (*Angular*, n.d.) Angular.io has a list of the properties:

* declarations:
  + This is described as components that are part of an NgModule (*Angular*, n.d.)
* exports:
  + More declarations that are in the “component templates of other NgModules” (*Angular*, n.d.)
* imports:
  + “Other modules whose exported classes are needed by component templates declared in this NgModule.” (*Angular*, n.d.)
* providers:
  + providers create services for this NgModule for global services, usable in all applications for the NgModule. (*Angular*, n.d.)
* bootstrap:
  + also called root component, it houses the application views (*Angular*, n.d.)

References:

*Angular*. (n.d.). Angular.io. Retrieved May 22, 2023, from https://angular.io/guide/architecture-modules