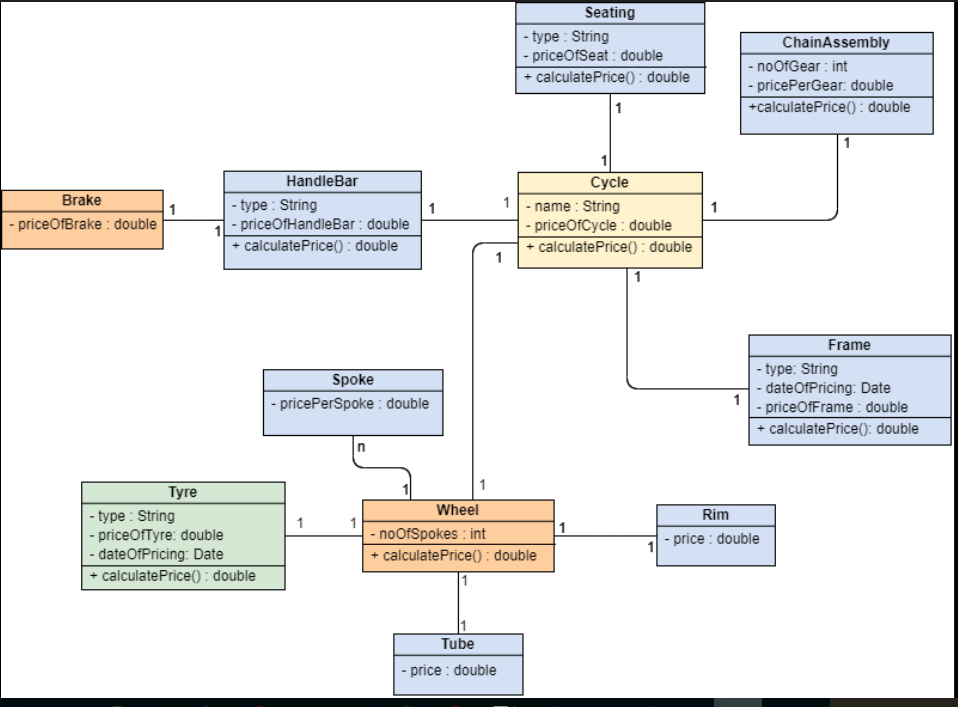
**Implementation of Cycle Price Engine**

**High level Class Diagram of Cycle**



**Pricing Engine Implementation:**

**Cycle Package**

High level components of cycle:

* ChainAssembly
* ChainAssembly Class: Implementation of calculating price for given ‘n’ gears
* Frame
* Frame Class: Implementation of Frame component of Cycle and calculating price for given type of frame and time sensitive
* HandleBar with Brake
* Brake Class: Implementation of Brake and initialize default brake price
* HandleBar Class: Implementation of HandleBar with Brake component of Cycle calculating price for given type of HandleBar along with Brake
* Seating
* Seating Class: Implementation of Seating component of Cycle and price calculation for given Seating type
* Wheel
* Rim Class: Implementation of Rim of wheel and initialize default Rim price
* Spoke Class: Implementation of Spoke of wheel and initialize default Spoke price
* Tube Class: Implementation of Tube of wheel and initialize default Tube price
* Tyre Class: Implementation of Tyre of wheel and calculate Tyre price based on type and time sensitive
* Wheel Class: Implementation of wheel component of Cycle and calculate the price of While for all the components of Wheel.

Cycle Class: Implementation of Cycle and calculates price of cycle for a given configuration.

CycleTest Class: Junit test for the Cycle class, **validatePrice()** validates cycle price with expected price for given input configuration. Also validates the price of components of Cycle.

Executing **validatePrice()** will perform Junit test for same.

Validation Class: Validates given configuration for cycle

**CycleScenario Package:**

CycleImplementation Class: Validates given configuration for Cycle, calculates price for given configuration and Date of pricing. Prints all outputs.

CycleImplementationTest Class: Junit testing for CycleImplementation Class, **calculatePrice()** method will calculate price of Cycle for given set of parameters of Cycle’s configurations and validate output with the given excepted output.

Executing **calculatePrice()** will perform Junit test for same.

Main Class: Prompts for input over command line argument and calls method to calculate cycle price for the given input configuration.

**ThreadImplementation Package:**

Main Class:

* Implements multi-threading, creates thread pool of 10
* Implements Producer class for producing configurations for 1000 Cycles as input and put into ArrayBlockingQueue.
* Implements Consumer class for consuming configurations for Cycle as input from ArayBlockingQueue and produce Price of Cycle as Output for 1000 Cycles.
* Executing **main()** method will create 10 Thread pool and calculate price for 1000 cycles

CycleThreadImpl Class: Cycle implementation of calculating price for given input and print necessary output for thread.