Saturday, November 21, 2020

Ruby 2 Study Guide

- Public methods can be accessed within a class definition, within a derived class, or outside of a class, and can be called with or without an implicit or explicit receiver. Protected methods can be accessed only within a class or derived classes and can be called both implicity or explicitly. Like Protected methods, Private methods can be accessed only within a class or derived classes, however, Private methods must be called implicitly on self except when using a private setter method.
- Modules unlike classes can't be instantiated. Modules contain methods that can be mixed into Ruby classes, and are a collection of methods that are shared among classes to keep our code DRY. Modules also take greater precedence over inherited classes.
- Big O is a relative representation of the time complexity of an algorithm. Big O describes time complexity of an algorithm given an arbitrarily large input. We use Big O to measure the efficiency of different approaches to a problem.
- Last In First Out is used to describe the operations used in the implementation of a Stack Abstract Data Type. Alternatively, First in First out , is used to describe the operations used in the implementation of a Queue ADT. Push/ Pop

-ADTs are more general than specific implementations of a data structure. Data structures are used to implement ADTs. A Data Structure is the way your data is stored specific to a programming language. Abstract data types are abstract ideas about how data can be interacted with. Abstract Data types also have specific ways you can interact with them. These ways they can be interacted with are sometimes called API.

- A few reasons to use Test Driven Development is that it allows us to make sure our code works, it increases our flexibility and reduces our fear of code so that we may feel free to refactor as we please because we can instantly check to see if code is broken, it makes collaboration amongst programmers easier when working amongst teams because we can be sure our code is performing its specified tasks, and lastly it produces documentation that we can refer to to know what our code does.
- Unit Tests are essentially testing as small of a unit of our code as possible. Generally testing methods in isolation. Mocks or doubles are generally used in unit tests and unit test are the vast majority of tests. Integration tests test how different units of our code work together to ensure that all parts of our code are working together well.

- A test double (also called a mock) is a fake object that we can use to create desired isolation. They are blank slate, waiting for us to add behaviors to it. We use doubles and mocks to be able to identify as narrowly as possible what's not working in our code. This is used for unit testing vs integration testing. Often used to imitate an instance of a class we have not implemented.
- Functionally speaking describe and context are identical. However according to general convention, describe is used to write specs for a specific method being tested. Context is for writing specs for testing a specific condition. Context are generally nested within a describe block.

- The basic workflow of Test Driven Development refers to the red green refactor. Writing the code for the specs, make sure all the specs we wrote then fail, then writing the code to pass the specs.