**MOOD METRICS**Diogo Correia 62475  
  
**Attribute Hiding Factor (AHF):**

So, the Attribute Hiding Factor (AHF) is like the undercover boss of your class attributes. It checks how well your class keeps its private stuff, like instance variables, away from the nosy external classes. A top-notch AHF, aiming for that cool 100%, means your class is the master of disguise, hiding its attributes like a pro. It's all about encapsulation, making sure your code stays tidy and easy to build upon.

**Attribute Inheritance Factor (AIF):**

Alright, let's talk about the Attribute Inheritance Factor (AIF). This one's all about the family ties between parent and child classes. A high AIF, cruising up to 48%, suggests a bit of a family drama with lots of attributes being passed down. But hold on, too much drama isn't always good. Keeping AIF on the down-low is the trick, lessening the dependency between classes for a smoother coding experience.

**Coupling Factor (CF):**

Now, the Coupling Factor (CF) is like the relationship status between your classes. A low CF is like saying, "We're just hanging out, no strings attached." Loose coupling means easy maintenance and reusable code – a coder's dream. But beware, a high CF screams "It's complicated," signaling tightly woven classes that could make changes a headache. Keeping it low is the key to a drama-free coding life.

**Method Hiding Factor (MHF):**

Picture the Method Hiding Factor (MHF) as your class's security guard for methods. A high MHF, hitting that sweet spot between 8% and 25%, means your methods are VIPs protected from unauthorized access. On the flip side, a low MHF is like leaving your methods out in the open – not the best move. Striking that balance is crucial, ensuring a safe yet functional method party.

**Method Inheritance Factor (MIF):**

Let's chat about the Method Inheritance Factor (MIF). It's the measure of how much your methods like to travel from parent to child classes. High MIF can bring in some complexity and dependencies between classes. Going for a lower MIF is like keeping things chill – less inheritance, less drama. It's all about finding that sweet spot for a codebase that's easy on the eyes.

**Polymorphism Factor (PF):**

Lastly, the Polymorphism Factor (PF) is like the cool factor of your code. Embracing polymorphism is like letting different classes party together, and that's awesome for flexibility and extensibility. But hold up, too much polymorphic action can make your code a bit confusing. So, finding that sweet spot between clear code and polymorphic fun is the name of the game. Keep it cool, not chaotic.