Mind Map: When to Use Getter and Setter Methods

Introduction

- This mind map summarizes the key points on when and how to use getter and setter methods.
- The sections cover common issues, scope, impact on object-oriented principles, and alternatives.

1. Issues with Getter and Setter

- Breaks encapsulation in object-oriented programming by exposing internal properties.
- Turns objects into 'passive data,' which can lead to a procedural approach rather than true object-oriented design.

2. Scope of Using Getter and Setter

- Only use when necessary, such as when there's no better way to expose certain data externally.
- Useful for creating flexible and extensible APIs, but should be used sparingly.

3. Impact on Polymorphism and Encapsulation

- May reduce polymorphism and encapsulation within the object.
- External classes may need to know about an object's internal implementation, breaking the object-oriented model.

4. Alternatives to Getter and Setter

- Use specific action methods instead of directly calling getter and setter methods.
- Encourage objects to manage and manipulate their data independently to prevent exposing internal details.