Dylan Perkins

April 3, 2023

Week 5 Research

Question: What are the four pillars of Object-Oriented Programming? Explain each pillar.

Answer: There are four main pillars at the foundation of OOP: Abstraction, Encapsulation, Inheritance, and Polymorphism. Abstraction refers to simplifying complex systems by breaking them down into smaller, more manageable pieces. Encapsulation is all about protecting data within objects by hiding internal workings from external elements. Inheritance allows subclasses to derive properties and behaviors from parent classes. And finally, Polymorphism enables multiple objects to take on different forms and behave in unique ways while still adhering to the same interface. Each of these pillars plays a crucial role in creating flexible, reusable, and organized code, making OOP an incredibly powerful and popular paradigm in software development.

Source: https://www.codingninjas.com/codestudio/library/four-pillars-of-oops-in-java#:~:text=OOPS%20moves%20around%20its%20four,Polymorphism%2C%20Abstraction%2C%20and%20Encapsulation.

Question: What are the differences between abstract classes and interfaces? When should you use one over the other?

Answer: An abstract class is a class that cannot be instantiated on its own, instead it serves as a base for derived classes. It contains one or more abstract methods, meaning methods that are declared but not defined, and concrete methods, meaning methods that are fully defined. Abstract classes can also contain member variables and constructors.

On the other hand, an interface is a contract that specifies a set of methods, properties and events that a class must implement. It does not provide any implementation of these members, only the signature.

So, when should you use one over the other? Well, you should use an abstract class when you want to provide a default implementation of some methods, while leaving others to be implemented by derived classes. On the other hand, you should use an interface when you want to ensure that a certain class implements some specific methods, but you don't care about the implementation itself.

Source: https://www.geeksforgeeks.org/difference-between-abstract-class-and-interface-in-java/#