

Activity 18-1: Design Patterns Review

A. What are design patterns? (10 min)

Discuss the following statements and come to an agreement about whether each statement is true or false.

1. A software design pattern is a general, reusable solution to a problem that commonly occurs in a given context when designing software.
2. Design patterns typically show relationships and interactions between classes or objects.
3. In a software development project, which design patterns to use is decided early on in the development process.
4. Design patterns formalize best practices for solving common programming problems.
5. Design patterns typically make code less complex.
6. Design patterns typically make code less flexible.
7. Code that solves a problem using a design pattern will be easy to plug in and reuse in other software projects to solve similar problems.

B. Design pattern roundup (20 min)

The leftmost column of the table below has descriptions of different design patterns, and the rightmost column describes example contexts when design patterns might be used. Match the descriptions on the left and the right to the column in the middle.

Command - Encapsulates information to perform an action so that the system can invoke the action without knowing anything specifics about the it.

- Useful for abstracting away details about the specific methods invoked by toolbar buttons or menu items so that their content can be easily changed.

Template Method Pattern Strategy Pattern Factory Pattern Facade Pattern Mediator Pattern Null Object Pattern Singleton Pattern Observer Pattern Adapter Pattern Proxy Pattern Bridge Pattern Visitor Pattern