Welcome!

- We'll start in a moment :)
- We may record tonight's event and plan to take screenshots for social media.
  - ***If you want to remain anonymous***, change your name & keep video off.
- We'll introduce the hosts and break in-between for Q/A.
- We will make some time for Q&A at the end of the presentation as well.
- Online event best practices:
  - Mute yourself when you aren't talking.
  - We want the session to be interactive.
  - Feel free to unmute and ask questions in the middle of the presentation.
  - Turn on your video if you feel comfortable

Check out:

- Technical Tracks
- Check our Digital Events
- Get updates – join the Digital mailing list
- Give us your feedback – take the Survey

WOMEN WHO CODE

WWCode Digital + **Backend**

Backend Study Group

May 20, 2021

WOMEN WHO CODE

Introduction & Agenda

- Welcome from WWCode!
- Our mission: Inspiring women to excel in technology careers.
- Our vision: A world where women are representative as technical executives, founders, VCs, board members and software engineers.

Prachi Shah
Senior Software Engineer @ Metromile

- **Software Design Patterns [Part 2 of 5]**
  - Creational Design Patterns [4/22]
  - **Structural Design Patterns [5/20]**
  - Behavioral Design Patterns [6/13]
  - Anti-patterns [6/17]
  - Interview Questions [7/1]
- What is Backend Engineering?
- Software Design
- Structural Design patterns

WOMEN WHO CODE

Backend Engineering

- What is Backend Engineering?
- Design, build and maintain server-side web applications.
- Concepts: Client-server architecture, API, micro-service, database engineering, distributed systems, storage, performance, deployment, availability, monitoring, etc.

Software Design

- Defining the architecture, modules, interfaces and data.
- Solve a problem or build a product.
- Define the input, output, business rules, data schema.
- Design patterns solve common problems.
- 3 Types:
  - UI design: Data visualization and presentation.
  - Data design: Data representation and storage.
  - Process design: Validation, manipulation and storage of data.

WOMEN WHO CODE

Backend Engineering

Design Patterns

- Set of template solutions that can be reused.
- Improved code maintainability, reusability and scaling.
- Leverages Object-oriented programming (OOP) principles for flexible and maintainable designs.
- Shared pattern vocabulary.
- Not a library or framework, but recommendations for code structuring and problem solving.
- Adapt a pattern and improve upon it to fit application needs.
- Defines relationship between objects, loosely coupled objects, secure code.

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	Factory Method	Adapter	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Proxy	Chain of Responsibility Command Iterator Mediator Memento Flyweight Observer State Strategy Visitor

WOMEN WHO CODE

Backend Engineering

Types of Design Patterns

- **Creational**:
  - Initialize a class and instantiate the objects.
  - Decoupled from implementing system.
  - Singleton, Factory, Builder.
  - Abstract Factory, Prototype.
- **Structural**:
  - Class structure and composition.
  - Increase code reusability and functionality.
  - Create large objects relationships.
  - Adapter, Facade, Decorator, Bridge, Composite, Flyweight, Proxy.
- **Behavioural**:
  - Relationship and communication between different classes.
  - Observer, Strategy, Iterator, etc.

Creational	Structural	Behavioral
• Factory Method	• Adapter	• Interperter
• Abstract Factory • Builder • Prototype • Singleton	• Adapter • Bridge • Composite • Decorator • Facade • Flyweight • Proxy	• Chain of Responsibility • Command • Iterator • Mediator • Momento • Observer • State • Strategy • Visitor

WOMEN WHO CODE

Backend Engineering

- What are design patterns?
- What is creational design pattern?
- What is structural design pattern?
- Can you give examples?

You can unmute and talk or use the chat.

WOMEN WHO CODE

Backend Engineering

Object-Oriented Programming Concepts:

- **Inheritance and *super()***: subclass inherits methods and attributes of superclass.
  super() inside subclass constructor to pass attributes.
- **Polymorphism**: same method but different behavior.
  - Overloading: Same name method but different parameters.
  - Overriding: Method signature (name & parameters) same in subclass and superclass.
- **Abstract class**:
  - Has *abstract* and concrete methods. Declare vs. Define variable and method.
  - Objects that extend (one) class have same/default behavior & can be overridden.
- **Interface**:
  - Has abstract methods only. Uses *implements* keyword.
  - Polymorphic behavior for objects that implement Interface(s).

WOMEN WHO CODE

Backend Engineering

Object-Oriented Programming Concepts:

- **Composition**:
  - Subclass cannot exists without superclass (conceptually). Strong association.
  - *Tree* superclass has *branch*, *leaves*, *fruit* subclasses.
- **Aggregation**:
  - Subclass can exists without superclass (conceptually). Weak association.
  - *Vehicle* superclass has *driver* subclass.
- ***is-a* and *has-a***:
  - is-a: Inheritance where subclass is-a superclass. *Mango* is a *Fruit.*
  - has-a: Composition where an object has-a another object. *Bookshelf* has *Book.*
- **Unified modeling language (UML) diagram**: Visualize design of a software.

Backend Engineering

- What are the OOP concepts?
- Can you give examples?

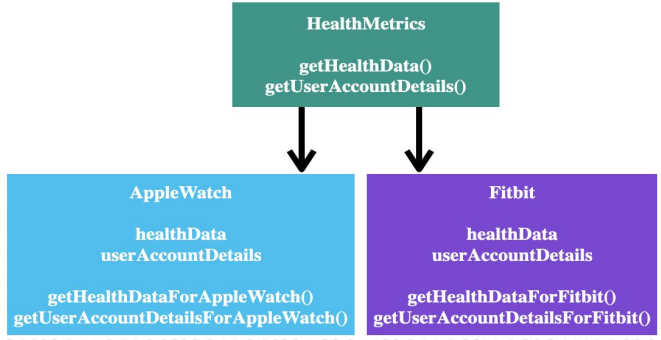
You can unmute and talk or use the chat.

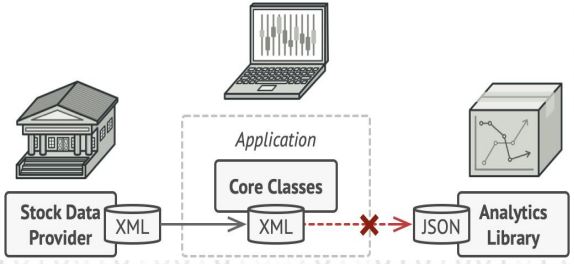
WOMEN WHO CODE

Backend Engineering

Structural Design Patterns:

- **Adapter:**
  - Wrapper pattern.
  - Incompatible objects can interact.
  - Object adapts to interface of another object.
  - Reusability of functionality.
  - Seperate the interface from business logic.
  - New adapters can be introduced for different client integrations.
  - Adapter:
    - Object that connects two different interfaces.
    - Wraps an object to hide the implementation complexity.
    - Object can use the interface, to call adapter methods.
  - Example: Connect your phone to Alexa, Fitbit, Apple Watch. APIs with XML/JSON
  - Code example.





WOMEN WHO CODE

Backend Engineering

- What is an Adapter design pattern?
- Can you give examples?

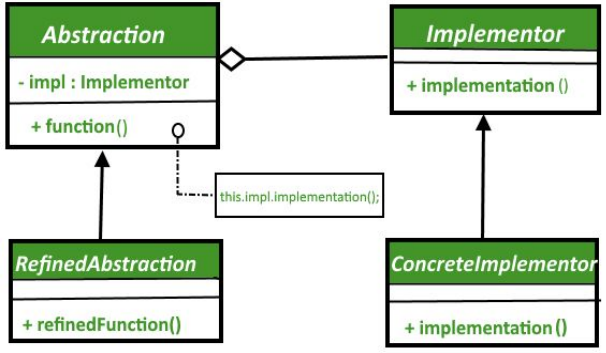
You can unmute and talk or use the chat.

WOMEN WHO CODE

Backend Engineering

Structural Design Patterns:

- **Bridge:**
  - Separate abstraction from implementation.
  - Independent development, loosely-coupled, hierarchical and hide details.
  - Client accesses abstraction, agnostic of implementation.
  - Abstraction:
    - Interface declare operations and delegates.
    - References the implementation.
    - *abstract* class and concrete class.
  - Implementor:
    - Operations are implemented.
    - *interface* and concrete implementor class that implements the interface.
  - Example: Lyft app has *driver* login and *rider* login.
  - Code example.



WOMEN WHO CODE

Backend Engineering

- What is a Bridge design pattern?
- Can you give examples?

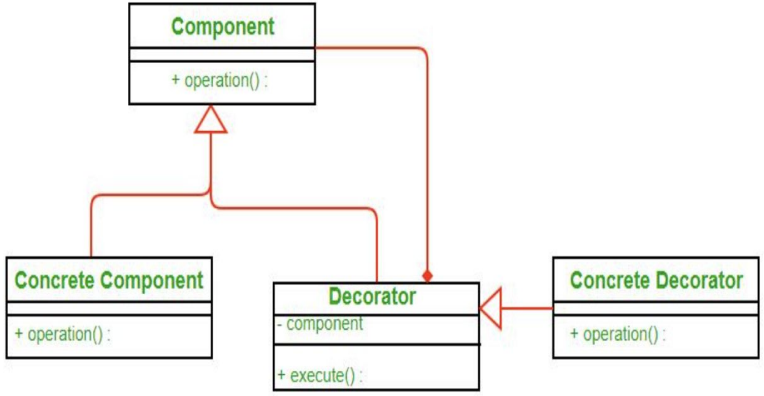
You can unmute and talk or use the chat.

WOMEN WHO CODE

Backend Engineering

Structural Design Patterns:

- **Decorator:**
  - Modify an object's behavior at runtime without modifying the structure.
  - Does not affect other object instances.
  - Removes need for subclassing, therefore more flexible than inheritance.
  - Extendible and easy to maintain code.
  - Decorator:
    - Class that encapsulates concrete class to provide modified functionality.
    - Wrapper linked to a target class.
    - Implements the same *interface* as the target class.
  - Example: Java IO classes like FileReader.
  - Code example.



WOMEN WHO CODE

Backend Engineering

- What is a Decorator design pattern?
- Can you give examples?

You can unmute and talk or use the chat.

WOMEN WHO CODE

Backend Engineering

Summary:

- Structural design patterns: Focus on class structure and composition.
- Adapter design pattern: Incompatible objects can interact.
- Bridge design pattern: Separate abstraction from implementation.
- Decorator design pattern: Modify an object's behavior at runtime without modifying the structure.

NEXT SESSION on 6-3-2021: Behavioral design patterns.

You can unmute and talk or use the chat.

WOMEN WHO CODE

Backend Study Group

- WWCode Presentation and Demo
- WWCode YouTube channel:
  - March 25, 2021 session recording: Backend Engineering
  - April 8, 2021 session recording: Java Microservice and REST API Demo
  - April 22, 2021 session recording: Creational Design Patterns
- **Resources**:
  - Software design pattern
  - Design Patterns in Java
  - Design patterns
  - Design Patterns
  - Head First Design Patterns book

You can unmute and talk or use the chat.

WOMEN WHO CODE