20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by «Umashankar kavya»
«cb.en.u4cys21082»
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

Outline

1. «Object-Oriented»
2. «Paradigm 1 - Swift»
3. «Concurrent»
4. «Paradigm 2 - GO»
5. Comparison and Discussions
6. Bibliography

What Is Object-Oriented - Swift And Concurrent - GO

Object-Oriented Programming(OOP) is a programming paradigm based on the conception of objects and classes, inheritance, encapsulation, abstraction, plymorphism.

Swift wift is a programming language developed by Apple. It is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C code written for Apple products.

Swift's goal is to create the best language for uses varying from desktop and mobile apps, systems programming and scaling up to cloud services.

Go provides channels that you can use for bidirectional communication between goroutines means that one goroutine will send a message and the other will read it. Sends and receives are blocking. Code execution will be stopped until the write and read are done successfully.

The primary goals of concurrency in Go are to provide a simple and efficient model for concurrent programming, particularly in the context of building scalable and concurrent systems.

Implementation of Object-Oriented - Swift

Object-oriented programming (OOP) in Swift is implemented through the following key features:

1) Classes and Object: Swift supports for creating class and object. Class can be created using 'Class' keyword and objects are the instance of these classes.

2) Encapsulation: Swift supports encapsulation. Generally encapsulation allows you to combine data and methods that work with class and hide the implementation details from user.

here in swift also encapsulation works for data hiding. We can achieve that using Access modifiers:

'private': Means that access to an attribute can only be accessed within an object.

'internal' : Anyone within the project can access the attribute.

'public' : Means that access is available within the workspace, mainly used for frameworks.

3) Inheritance: Swift supports inheritance.

It allows to inherit the properties and methods from one class to another.

We use 'super' keyword to call the superclass method or to access their properties and even we use 'class' keyword.

Implementation of Object-Oriented - Swift

4) Polymorphism: Swift supports polymorphism. child class can override an attribute, using the override modifier.

It can be achieved by method overriding. Subclasses can provide a specific implementation of a method already defined in their superclass.

5) Abstraction: In Swift, there is no such thing as an abstract class, but a class that has no instances can be considered abstract. class can implement several interfaces, and one interface can be implemented by several classes.

Abstraction is achieved by defining classes with properties and methods that model real-world entities, hiding unnecessary details.

Implementation of Concurrent - GO

Concurrency in Go is implemented through goroutines and channels.

Goroutines are light weight threads of execution.

Channels establishes the communcation and synchronization between these goroutines.

Goroutines: 1. Creating Goroutines:

We can create new goroutine using 'go' keyword followed by function call

In go, as soon as the application terminates, along with it the goroutine also exits

Channels:

1. Creating channels:

It can be created using 'make' function.

Creating over unbuffered channel: An unbuffered channel has no capacity; it can hold only one value at a time.

Communication over an unbuffered channel involves both sending and receiving Goroutines synchronizing with each other.

Implementation of Concurrent - GO

Creating over buffered channel: A buffered channel has a specified capacity, allowing it to hold multiple values.

The sender is not blocked until it becomes full and it doesn't necessarily need a reader to complete the synchronization with every operation.

2. Bidirectional and unidirectional channels:

Bidirectional communication means that one goroutine will send a message and the other will read it. Sends and receives are blocking.

Code execution will be stopped until the write and read are done successfully

Comparison between SWIFT and GO

SWIFT:

1. Generally swift language is used for developing iOS, macOS, watchOS, and tvOS applications.Widely used in mobile apps, desktop applications, server-side development.

: 2. Swift does not have a separate library import to support functionalities like input/output or string handling.

3. Swift's object-oriented paradigm is familiar to developers with experience in languages like Java, C++, or Objective-C.

4. Swift is a versatile language with strong support for object-oriented programming, making it well-suited for a variety of application domains.

5. The syntax of Swift would be more familiar to web developers. Swift designers took ideas from other programming languages such as Objective-C, Rust, Haskell, Ruby, Python, C, and CLU.

Comparision between SWIFT and GO

GO:

1. Go is widely used for building scalable and concurrent systems, such as web servers, microservices, and distributed systems.

2. You don't have to compile your Go code to run it. It will be automatically compiled and run. A significant strength of Go is that it's minimalistic and fast., Monitoring and Logging Systems.

3.Go can be a powerful tool for web programming, microservices, or mobile development. In many use cases, Go web development has proved to be more rapid than Swift.

4. It is known for its efficiency in handling concurrency, making it a good choice for systems programming and cloud-native applications.

5. Go encourages the use of channels to communicate between Goroutines rather than shared memory, reducing the complexity of handling shared state and avoiding common concurrency issues.

Real World Applications of SWIFT and GO

SWIFT:

iOS App Development, macOS App Development, WatchOS App Development,

Server-Side Swift, Game Development, Augmented Reality (AR) Apps,

Health and Fitness Apps, Educational Apps.

GO:

Web Servers and APIs, Microservices Architecture,

Cloud Infrastructure, Container Orchestration,

Networking Tools, Data Processing and Pipelines, Financial Systems.

Reference Link

https://www.freecodecamp.org/news/concurrent-programming-in-go/

https://medium.com/macoclock/object-oriented-programming-in-swift-8e0a379b111a

https://www.educba.com/swift-vs-go/

https://medium.com/@asappstudio/key-features-of-swift-apples-programming-language-b2a226602ca6