Similarities of both structs and classes:

- Stored vars: Variables that store data associated with the instance.
- Computed vars: Variables that are calculated based on other properties of the struct or class.
- Constant lets: Constants that store data associated with the instance.
- Functions: Methods that can be called on the struct or class instance.
- Initializers: Methods that are used to initialize the struct or class instance.

Here is an example of a struct with stored vars, computed vars, constant lets, functions, and initializers:

Swift

```swift
struct Point {
    var x: Double
    var y: Double

    // Computed var
    var distanceFromOrigin: Double {
        return sqrt(x * x + y * y)
    }

    // Constant let
    let name: String

    // Function
    func move(by dx: Double, dy: Double) {
        x += dx
        y += dy
    }

    // Initializer
    init(x: Double, y: Double, name: String) {
        self.x = x
        self.y = y
        self.name = name
    }
}
```

Here is an example of a class with stored vars, computed vars, constant lets, functions, and initializers:

Swift

```swift
class User {
    var name: String
    var email: String

    // Computed var
    var fullName: String {
        return "\(name) \(email)"
    }

    // Constant let
    let id: Int

    // Function
    func changeEmail(to newEmail: String) {
        email = newEmail
    }

    // Initializer
    init(name: String, email: String, id: Int) {
        self.name = name
        self.email = email
        self.id = id
    }
}
```

As you can see, structs and classes are very similar in terms of the features they support. The main difference is that structs are value types and classes are reference types.