Основы Swift

Жизненный цикл объектов

Инициализатор

```
class View {
    var height: Float
    var width: Float

    init() {
        height = 0
        width = 0
    }

    deinit {
        print("Объект View удален")
    }
}
```

Краткая форма и деструктор

```
let view = View.init()
let anotherView = View()

var optionalView: View? = View()
optionalView = nil
```

Значение по умолчанию

```
class View {
    var height = 0.0
    var width = 0.0

    init() {
    }
}

let view = View()
```

Инициализатор по умолчанию

```
class View {
    var height = 0.0
    var width = 0.0
}

let view = View()
```

Memberwise инициализатор

```
struct User {
    var name: String
    var email: String
    var age: Int
}

var user = User(name: "Bob", email:
"bog@mail.com", age: 46)
```

Параметры
для инициализатора

```
class View {
    let height: Double
    let width: Double

    init(side: Double) {
        height = side
        width = side
    }
}

let view = View(side: 5)
```

Делегирование инициализаторов

```
struct Size {
    var width = 0.0
    var height = 0.0
}

struct Point {
    var x = 0.0
    var y = 0.0
}
```

```
struct Rect {
    var origin = Point()
    var size = Size()

    init() {}

    init(origin: Point, size: Size) {
        self.origin = origin
        self.size = size
    }

    init(center: Point, size: Size) {
        let originX = center.x -
        (size.width / 2)
        let originY = center.y -
        (size.height / 2)
        self.init(origin: Point(x: originX,
        y: originY), size: size)
}
}
```

Failable инициализатор

```
struct Size {
    var width = 0.0
    var height = 0.0

    init() {}

    init?(width: Double, height: Double) {
        guard width >= 0 && height >= 0 else {
            return nil
        }

        self.width = width
        self.height = height
    }
}

let size = Size(width: -10, height: 50)
size // nil
```

Инициализатор для перечислений

```
enum TemperatureUnit: Character {
    case kelvin = "K"
    case celsius = "C"
    case fahrenheit = "F"
}

let unknownUnit = TemperatureUnit(rawValue: "X")
if unknownUnit == nil {
    print("Неизвестная единица измерения")
}
```

Вызов Failable инициализатора

```
class MyClass {
    init?(someValue: Int) {
    }

    convenience init() {
        self.init(someValue: 5)!
    }
}
```

Замыкания в инициализаторах

```
class View {
    var frame: Rect = {
        let randomX =
        Double(arc4random_uniform(100))
        let randomY =
        Double(arc4random_uniform(100))
        let position = Point(x: randomX, y:
        randomY)
        return Rect(center: position, size:
        Size(width: 50.0, height: 50.0)!)
    }()
}
```