



Основы Swift

Определение

Generics

Перегруженный метод

```
func customDescription(_ arg: Int) -> String {  
    return "Аргумент: \(arg)"  
}
```

```
func customDescription(_ arg: (Double, Bool)) ->  
String {  
    return "Аргумент: \(arg)"  
}
```

```
func customDescription<PlaceholderType>(_  
arg: PlaceholderType) -> String {  
    return "There is our arg: \(arg)"  
}
```

Определение

```
func имяФункции<Плейсхолдер1,  
Плейсхолдер2>(параметр1: Плейсхолдер1,  
параметр2: Плейсхолдер2) -> Плейсхолдер2
```

Реализация стека

```
struct StackOfInteger {  
    var items = [Int]()  
    mutating func push(_ item: Int) {  
        items.append(item)  
    }  
    mutating func pop() -> Int {  
        return items.removeLast()  
    }  
}
```

Использование Any

```
struct StackOfAny {  
    var items = [Any]()  
    mutating func push(_ item: Any) {  
        items.append(item)  
    }  
    mutating func pop() -> Any {  
        return items.removeLast()  
    }  
}
```

```
let item = "Contained info"
```

```
var stackOfAny = StackOfAny(items: [item])  
let poppedAnyItem = stackOfAny.pop() // Any тип  
let stringValue: String = poppedAnyItem  
                           as! String // String тип
```

Дженерик параметры

```
struct StackOfGenerics<ContainedType> {  
    var items = [ContainedType]()  
    mutating func push(_ item: ContainedType) {  
        items.append(item)  
    }  
    mutating func pop() -> ContainedType {  
        return items.removeLast()  
    }  
}
```


Дженерик параметры

```
var stackOfGenerics = StackOfGenerics(items:  
                                     [item])  
let poppedGenericItem = stackOfGenerics.pop()  
                        // String тип  
let result = (poppedAnyItem as! String) +  
             poppedGenericItem
```

@_specialize

```
struct StackOfGenerics<ContainedType> {  
    var items = [ContainedType]()  
  
    @_specialize(where ContainedType == Int)  
    mutating func push(_ item: ContainedType) {  
        items.append(item)  
    }  
  
    @_specialize(where ContainedType == Int)  
    mutating func pop() -> ContainedType {  
        return items.removeLast()  
    }  
}
```