Table of Contents

1. Язык программирования Scala

Рисунок 1

1.1. Общая характеристика и история создания

«Scala(Рисунок 1) — мультипарадигмальный язык программирования, спроектированный кратким и типо безопасным для простого и быстрого создания компонентного программного обеспечения, сочетающий возможности функционального и объектно-ориентированного программирования.

Первые версии языка созданы в 2003 году коллективом лаборатории методов программирования Федеральной политехнической школы Лозанны под руководством Мартина Одерски, язык реализован для платформ Java и .Net. По мнению Джеймса Стрэчена (англ. James Strachan), создателя языка программирования Groovy, Scala может стать преемником языка Java.» [1]

1.2. Система типов и особенности архитектуры языка

«Scala - программы во многом похожи на Java-программы, и могут свободно взаимодействовать с Java-кодом. Язык включает единообразную объектную модель (Таблица 1) — в том смысле, что любое значение является объектом, а любая операция — вызовом метода. При этом является также функциональным языком в том смысле, что функции — это полноправные значения.

В Scala включены мощные и единообразные концепции абстракций как для типов, так и для значений. В частности, язык содержит гибкие симметричные конструкции примесей для композиции классов и типажей. Возможно позволяет производить декомпозицию объектов путём сравнения с образцом; образцы и выражения при этом были обобщены для поддержки естественной обработки XML-документов. В целом, эти конструкции позволяют легко

выражать самостоятельные компоненты, использующие библиотеки Scala, не пользуясь специальными языковыми конструкциями.» [2]

Таблица 1

Тип	Описание
Byte	8 битовое знаковое значение. Диапазон от -128 до 127
Short	16 битовое знаковое значение. Диапазон от -32768 to 32767
Int	32 битовое знаковое значение. Диапазон от -2147483648 to 2147483647
Long	64 битовое знаковое значение. Диапазон от -9223372036854775808 to 9223372036854775807
Float	32 битовое IEEE 754 число с плавающей точкой одинарной точности
Double	64 битовое IEEE 754 число с плавающей точкой двойной точности
Char	16 битовое беззнаковый символ Unicode. Диапазон от U+0000 to U+FFFF
String	Последовательность экземпляров Char

1.3. Примеры кода и интересные факты

«Программа, как и в Java, представляет собой класс. Это пример консольной программы, которая выводит строчку текста на экран.

```
object HelloWorld {
  def main(args: Array[String]) =
    println("Привет, МИР!")
}
```

Следующий простой пример программы написан на Java, Scala и C#, демонстрирующий некоторые различия в синтаксисе (постфиксная запись типов переменных, отсутствие специального синтаксиса для доступа к массивам). В этом примере описывается консольная программа, которая выводит все опции, переданные через командную строку. Опции начинаются с символа «-» (минус).” [3]

```
// Java:
```

```
 class PrintOptions {
    public static void main(String args[]) {
      System.out.println("Выбраны опции:");
      for (String arg: args) if (arg.startsWith("-"))
System.out.println(" " + arg.substring(1));
    }
 }
```

```
// Scala:
 object PrintOptions {
   def main(args: Array[String]) {
     println("Выбраны опции:")
     for (arg <- args if arg startsWith "-") println(" " + (arg
substring 1))
   }
 }
```

```
// В функциональном стиле Scala:
 object PrintOptions {
   def main(args: Array[String]) =
     println("Выбраны опции:" +: (args filter (_ startsWith "-") map ("
" + _.drop(1))) mkString "\n")
 }
// В функциональном стиле C#:
 class PrintOptions {
   static void Main(String[] args) {
     Console.WriteLine("Выбраны опции:" + args.Where(x =>
x.StartsWith("-")).Aggregate((r, x) => r + " " + x.Substring(1)));
   }}
```

```
// В функциональном стиле Java:
 class PrintOptions {
    public static void main(String[] args) {
      System.out.println("Выбраны опции:");
      Arrays.stream(args).filter(o -> o.startsWith("-")).forEach(o ->
System.out.println(" " + o.substring(1)));
    }
 }
```

2. Язык программирования Go

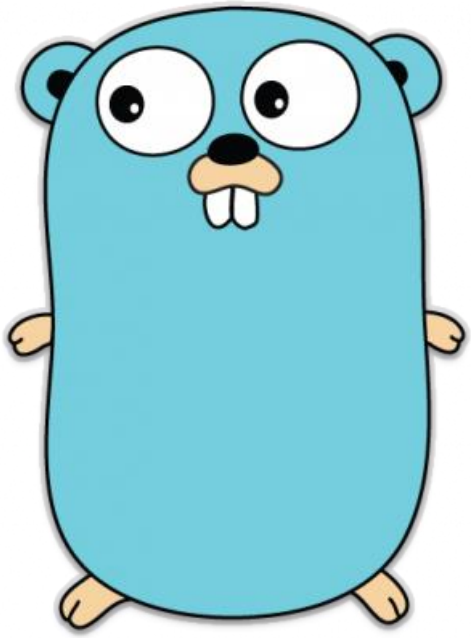

Рисунок 2

2.1 Общая характеристика и история создания

«Go (часто также Golang) 5 (Рисунок 2)— компилируемый многопоточный язык программирования, разработанный внутри компании Google. Первоначальная разработка Go началась в сентябре 2007 года, а его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон, занимавшиеся до этого проектом разработки операционной системы Inferno. Официально язык был представлен в ноябре 2009 года. На данный момент его поддержка осуществляется для операционных систем FreeBSD, OpenBSD, Linux, macOS, Windows, начиная с версии 1.3 в язык Go включена экспериментальная поддержка DragonFly BSD, Plan 9 и Solaris, начиная с версии 1.4 — поддержка платформы Android.» [4]

2.2 Система типов и особенности архитектуры языка

«Все данные, которые хранятся в памяти, по сути представляют просто набор битов. И именно тип данных определяет, как будут интерпретироваться эти данные и какие операции с ними можно производить. Язык Go является статически типизированным языком, то есть все используемые в программе данные имеют определенный тип.(Таблица 2)»[5]

«Язык Go не поддерживает типичный для большинства современных языков синтаксис структурной обработки исключений (блоки try-catch); авторы сочли, что его применение провоцирует программиста на игнорирование ошибок.

Модель многопоточности Go была создана на основе CSP Тони Хоара по типу предыдущих распараллеливаемых языков программирования Occam и Limbo, но также присутствуют такие особенности как Пи-исчисления и канальная передача.

Специальное ключевое слово для объявления класса в Go отсутствует, но для любого именованного типа, включая структуры и базовые типы вроде int, можно определить методы, так что в смысле ООП все такие типы являются классами. » [6]

Таблица 2

Тип	Описание
Byte	8 битовое знаковое значение. Диапазон от -128 до 127
Short	16 битовое знаковое значение. Диапазон от -32768 to 32767
Int	32 битовое знаковое значение. Диапазон от -2147483648 to 2147483647
Long	64 битовое знаковое значение. Диапазон от -9223372036854775808 to 9223372036854775807
Float	32 битовое IEEE 754 число с плавающей точкой одинарной точности
Double	64 битовое IEEE 754 число с плавающей точкой двойной точности
Char	16 битовое беззнаковый символ Unicode. Диапазон от U+0000 to U+FFFF
String	Последовательность экземпляров Char

2.3 Примеры кода и интересные факты

«Программа, как и в Java, представляет собой класс. Это пример консольной программы, которая выводит строчку текста на экран.

```
Package main
Import "fmt"
```

```
 func main() {
   fmt.Println("Привет, МИР!")
}
```

```
object PrintOptions {
   def main(args: Array[String]) {
     println("Выбраны опции:")
     for (arg <- args if arg startsWith "-") println(" " + (arg
substring 1))
   }
}
```

3. Язык программирования Swift

3.1 Общая характеристика и история создания

Рисунок 3

«Swift (Рисунок 3) — открытый мультипарадигмальный компилируемый язык программирования общего назначения. Создан компанией Apple в первую очередь для разработчиков iOS и macOS. Swift работает с фреймворками Cocoa и Cocoa Touch и совместим с основной кодовой базой Apple, написанной на Objective-C. Swift задумывался как более лёгкий для чтения и устойчивый к ошибкам программиста язык, нежели предшествовавший ему Objective-C[7]. Программы на Swift компилируются при помощи LLVM, входящей в интегрированную среду разработки Xcode 6 и выше. Swift может использовать рантайм Objective-C, что делает возможным использование обоих языков (а также C) в рамках одной программы.

»[7]

3.2 Система подтипов и особенности архитектуры языка

«Типы данных (Таблица 3), которые обычно считаются основными или примитивными в других языках, такие как типы, которые представляют цифры, символы и строки-фактически именованные типы, определены и реализованы в стандартной библиотеке Swift с использованием структур. Так как это именованные типы, вы можете расширить их поведение для удовлетворения потребностей вашей программы, используя объявление расширения .» [8]

Таблица 3

Тип	Описание
Integer	i8, u8, i16, u16, i32, u32, i64, u64, а также isize и usize, имеющие размер указателя на данной платформе. Примеры значений: -5i8, 0x400u16, 0o100i16, 20_922_789_888u64, b'*' (u8 char), 42 (по предположению), 0xfff_fc00usize
Float	f32, f64. Примеры: 3.14f32, 6.0221e23f64
Bool	true, false
String	Тип, представляющий символ Unicode (внутреннее представление данных как u32). Примеры значений: '*', '\n', '\x7f', '\u{CA0}'

3.1. Примеры кода и интересные факты

```
Print(“Hello, World”)
Let people = [“Anna”:62, “Beto”:8]
For (name,age)in people {
    Print(“\(name) is \(age)”)
}[7]
```

Bibliography

[1] "Scala (Programming Language)"
[2] "Wikipedia (Ключевые аспекты языка),"
[3] "Wikipedia (Примеры).”
[4] "Wikipedia Go”
[5] "Metanit Go"
[6] "Wikipedia Go (Примеры)”
[7] "Wikipedia Swift.”
[8] "Swift Book”

Опрос