

**Примерные оценочные материалы, применяемые при проведении  
промежуточной аттестации по дисциплине (модулю)**

**«Микросервисная архитектура»**

Примерный перечень вопросов

1) Ближайший конкурент Go по скорости является:

- a. Java
- b. C++
- c. COBOL
- d. Си**
- e. Таковых нет

2) Что выведет код ниже?

```
var happiness = 0.04
var isHappy = true
if happiness >= 0.5 || isHappy {
    fmt.Println("Happy ")
}

fmt.Println(":)")
```

- a. :)
- b. Happy
- c. Happy :)**
- d. Будет выведена ошибка

3) Верно ли ниже указан импорт пакетов?

```
import ("fmt"; "os"; "net/http")
```

- a. Неверно указаны названия модулей
- b. Нужно прописывать через запятую
- c. Круглые скобки не требуются
- d. Код указан верно**
- e. Все варианты ошибок верны

- 4) Где верно создана переменная?
- a. `var age int = 12`
  - b. `var age = 12`
  - c. `var isDone bool = true`
  - d. Нет верного варианта
  - e. **Все варианты верные**
- 5) Что из следующего заставляет цикл пропустить оставшуюся часть своего тела и сразу вернуться к его условию?
- a. `break`
  - b. **`continue`**
  - c. `goto`
  - d. Ни один из вышеперечисленных
- 6) Термин `gvalue` относится к значению данных, которое хранится по некоторому адресу в памяти?
- a. **Да**
  - b. Нет
  - c. Это должны решать специалисты
- 7) Что верно относительно `map` в Go?
- a. Go предоставляет еще одну важную карту типа данных, которая отображает уникальные ключи к значениям
  - b. Ключ – это объект, который вы используете для получения значения позднее
  - c. **Оба вышеупомянутых**
  - d. Ни один из вышеперечисленных
- 8) Что из следующего верно об инструкции `break` в Go?
- a. Когда оператор `break` встречается внутри цикла, цикл немедленно завершается, и программный контроль возобновляется при следующем утверждении, следующем за циклом
  - b. Его можно использовать для завершения дела в инструкции `switch`

**с. Оба вышеупомянутых**

**d. Ни один из вышеперечисленных**

9) В Go есть привычные массивы, но есть и другое понятие — слайс байт.

Слайсы аналогичны массивами в других языках программирования, но со своими особенностями. Строка в Go — это тоже слайс байт. А вот для работы с одним символом используется тип...

a. char

**b. rune**

c. string

d. int

10) В Go есть горутина — goroutine. С помощью goroutine вы можете конкурентно выполнять функции в том же адресном пространстве.

Что-то вроде запуска функции в «легковесном» треде. С версии 1.4 запуск одной goroutine занимает на старте...

**a. 2048 байт**

b. 512 байт

c. 8192 байта

d. 1024 байта

11) Go славится своей простотой, но для некоторых базовых вещей отсутствует элегантное решение. Например, для работы со слайсами не добавлены «базовые» операции над ним. Поэтому, чтобы удалить второй элемент в этом слайсе, нужно использовать код...

```
sl := []int{1, 3, 4}
n := 1
```

**a. res := append(sl[:n], sl[n+1:]...)**

b. sl.delete(n)

c. sl[n] = nil

d. Удалить второй элемент нельзя

- 12) Перед вами листинг простой программы, которая печатает результаты в несколько потоков. Какое максимальное количество потоков может быть запущено при выполнении этой программы?

```
import (
    "fmt"
    "time"
)

func main() {
    for i := 1; i <= 3; i++ {
        go worker(i)
    }

    time.Sleep(1 * time.Second)
}

func worker(id int) {
    fmt.Println(id)
}
```

- a. 2
  - b. 3
  - c. 4**
  - d. 10
- 13) Что необходимо добавить вместо ABC, чтобы на любой запрос отвечать “Hello, RUT!”?

```
package main

import(
    "fmt"
    "net/http"
)

func main() {
    mux := http.NewServeMux()
    mux.HandleFunc("/", func(w http.ResponseWriter, req *http.Request)
    {
```

```

    fmt.Fprintf(w, "Hello, RUT!")
})

// ABC
}

a. http.Wait(":8000", mux)
b. http.Rub(":8000", mux)
c. http.ListenAndServe(":8000", mux)
d. http.Listen(8000) http.Serve(mux)

```

14) Что напечатает программа?

```

package main

var n int = 5

func main() {
    var x = byte(10 << n / 10)
    var w = 10 << n
    var y = byte(w / 10)
    var z = byte(w) / 10
    println(x, y, z)
}

```

- a. 32 32 32
- b. 6 32 32
- c. 6 32 6**
- d. 32 32 6

15) Каким из следующих методов является метод по умолчанию для передачи параметров в Go?

- a. Передача по значению**
- b. Передача по ссылке
- c. Оба вышеупомянутых
- d. Ни один из вышеперечисленных