

1. Suppose you want to start a goroutine which executes a function called **test1()**. What code would create this goroutine?

1 / 1 балл

- ☐ **test1()** go
- ☐ start test1()
- ☐ goroutine test1()
- ☒ go test1()

✓ Правильно  
Correct!

2. When does a goroutine complete?

1 / 1 балл

- I. When its code completes.
- II. When all goroutines complete.
- III. When the main goroutine completes.

- ☐ I and II, NOT III.
- ☒ I and III, NOT II.
- ☐ I, II, and III.
- ☐ I only.

3. Synchronization is useful for what purpose?

1 / 1 балл

- I. Restrict illegal interleavings.
- II. Force events in different goroutines to occur in sequence.
- III. Allow a goroutine to continue to execute after the main goroutine has completed.

- ☐ I, II, and III.
- ☐ I only.
- ☐ I and III, NOT II.
- ☒ I and II, NOT III.

✓ Правильно  
Correct!

4. If a goroutine g1 is using a WaitGroup wg to wait until another goroutine g2 completes a task, what method of the the WaitGroup should be called when g2 has finished the task?

1 / 1 балл

- ☒ wg.Done()
- ☐ wg.End()
- ☐ wg.Finished()

—

5. If a goroutine g1 is using a WaitGroup wg to wait until another goroutine g2 completes a task, what method of the the WaitGroup should be called *before* g2 starts its task?

1 / 1 балл

- ☐ wg.Fork()
- ☐ wg.Start()
- ☒ wg.Add()
- ☐ wg.Begin()

✓ Правильно  
Correct!

6. How might you write code to allow a goroutine to receive data from a channel c?

1 / 1 балл

- ☐ x <- c
- ☒ x = <- c
- ☐ x = c
- ☐ x <-- c

✓ Правильно  
Correct!

7. What is the difference between a buffered channel and an unbuffered channel?

1

- ☒ A buffered channel can hold multiple objects until they are read. An unbuffered channel cannot.
- ☐ A buffered channel delays the transmission of data. An unbuffered channel does not.
- ☐ A buffered channel delays the reception of data. An unbuffered channel does not.
- ☐ A buffered channel can communicate between more than 2 goroutines. An unbuffered channel cannot.

✓ Правильно  
Correct!