

1.

```
import UIKit

func +(l: CGPoint, r: CGPoint) -> CGPoint {
    return CGPoint(x: l.x + r.x, y: l.y + r.y)
}

var rect = CGRect(x: 0, y: 0, width: 320, height: 480)
rect.origin + CGPoint(x: 10, y: 10)
```

Какое значение будет в rect?

1. {x 10, y 10, width 320, height 480}
2. {x 0, y 0, width 320, height 480}
3. Compiler error

2.

```
import UIKit

extension CGSize {
    mutating func scale(by f: CGFloat) {
        width *= f
        height *= f
    }
}

let s = CGSize(width: 100, height: 100)
s.scale(by: 2)
```

Какое значение будет в s?

1. {width 100, height 100}
2. {width 200, height 200}
3. Compiler error

3.

```
import UIKit
class A {
    var b: B
    init(b: B) {
        self.b = b
    }
    deinit {
        print("A")
    }
}

class B {
    weak var a: A?
    deinit {
        print("B")
    }
}

var b: B? = B()
var a: A? = A(b: b!)
b?.a = a
b = nil
```

Что будет выведено на экран?

1. B
2. BA
3. AB
4. Ничего

4.

```
import UIKit
func counter() -> (Int) -> String {
    var total = 0
    func add(_ x: Int) -> String {
        total += x
        return "\(total)"
    }
    return add
}

let f = counter()
print(f(1))
let g = counter()
print(g(2))
print(f(3))
```

Что будет выведено в последней строке?

1. 6
2. 5
3. 4
4. 0

5.

```
import UIKit

var dictWithNils: [String: Int?] = [
    "one": 1,
    "two": 2,
    "none": nil
]

dictWithNils["two"] = nil
dictWithNils["none"] = nil
let result = dictWithNils.count
print(result)
```

Что будет выведено на экран?

- 1.
- 3
- 2
- Compiler error

6.

Выберите неверное утверждение:

1. Проект включает в себя все ресурсы и файлы
2. Таргет содержит информацию о сортировке и группировке файлов
3. Проект хранит настройки и описание таргетов
4. Таргет содержит настройки и фазы сборки

7.

Выберите неверное утверждение:

1. Проект хранит настройки и описание таргетов
2. Проект содержит информацию о сортировке и группировке файлов
3. Проект содержит настройки и фазы сборки
4. Таргет хранит список файлов для сборки

8.

Какова роль UIApplicationMain?

1. Принимает пользовательские события, передает их UI компонентам
2. Обрабатывает события жизненного цикла приложения
3. Запускает фоновые задачи
4. Создает объект приложения, настраивает цикл обработки событий

9.

Выберите неверное утверждение о UIApplication:

1. Принимает пользовательские события, передает их UI компонентам
2. Открывает ссылки на внешние источники или в другие приложения (URL scheme)
3. Запускает фоновые задачи
4. Создает объект приложения, настраивает цикл обработки событий

10.

Выберите верное утверждение о UIApplicationDelegate:

1. Принимает пользовательские события, передает их UI компонентам
2. Обрабатывает события жизненного цикла приложения
3. Запускает фоновые задачи
4. Создает объект приложения, настраивает цикл обработки событий

11.

Что из перечисленного вызывается раньше?

1. UIApplicationMain()
2. application:didFinishLaunchingWithOptions()
3. viewDidLoad()
4. application:willFinishLaunchingWithOptions()

12.

Выберите неверное утверждение об Objective-C:

1. Динамическая типизация
2. Доступен method swizzling
3. Поддерживает namespace (вложенные классы)
4. Совместимость с языком C

13.

Выберите верное утверждение об Objective-C:

1. Строгая типизация
2. Доступен method swizzling
3. Поддерживает namespace (вложенные классы)
4. Доступен playground

14.

Выберите неверное утверждение о Swift:

1. Динамическая типизация
2. Безопаснее благодаря опционалам
3. Поддерживает namespace (вложенные классы)
4. Доступен playground

15.

Выберите верное утверждение о Swift:

1. Динамическая типизация
2. Доступен method swizzling
3. Поддерживает namespace (вложенные классы)
4. Совместимость с языком C