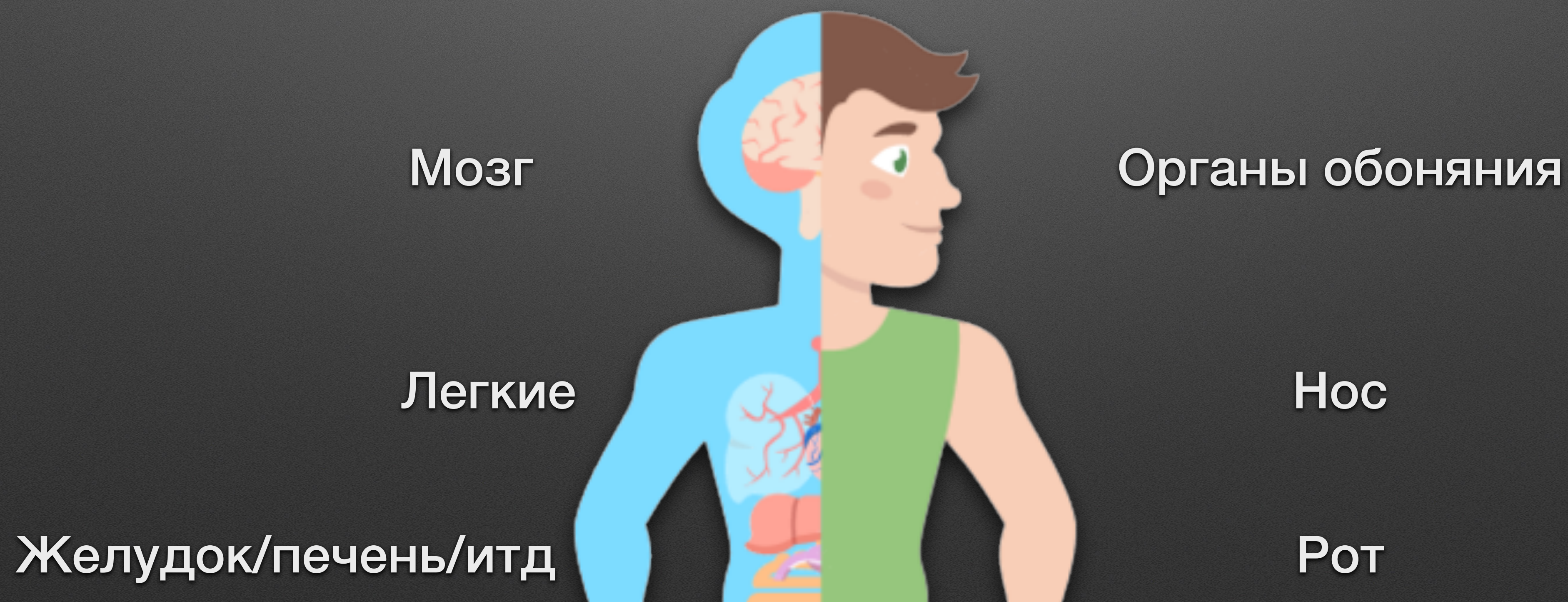


Инкапсуляция

ООП
Пахаев Х.Х.

Что такое инкапсуляция?

Инкапсуляция означает скрывание деталей реализации и предоставление интерфейса для взаимодействия с объектом.



Доступ к данным класса на C++

Доступ к данным класса контролируется с использованием модификаторов доступа: `public`, `private` и `protected`.

- `public` члены класса доступны извне и используются для взаимодействия с объектом.
- `private` члены класса скрыты и доступны только внутри класса.
- `protected` члены класса доступны только внутри данного класса и внутри дочерних классов

Доступ к данным `public`

- Члены класса, объявленные как `public`, являются общедоступными, что означает, что они могут быть использованы из любой части программы, включая код за пределами класса.
- Публичные члены образуют интерфейс класса, предоставляя внешнему коду методы доступа к данным и функциональности класса.
- Этот модификатор используется для элементов, которые должны быть доступны для использования извне класса.

Доступ к данным `private`

- Члены класса, объявленные как `private`, являются приватными и доступны только внутри самого класса.
- Приватные члены скрыты от внешнего кода и не могут быть использованы напрямую.
- Этот модификатор обеспечивает инкапсуляцию данных, скрывая детали реализации и предотвращая несанкционированный доступ.

Доступ к данным `protected`

- Члены класса, объявленные как `protected`, похожи на приватные члены, но они доступны также в производных классах (наследниках).
- Это модификатор, который используется в наследовании, чтобы предоставить доступ к базовым классам для их производных классов.
- Производные классы могут получить доступ к защищенным членам базового класса.

Порядок объявления модификаторов доступа

В C++, порядок объявления модификаторов доступа `public`, `private` и `protected` не имеет значения. Вы можете объявить их в любом порядке внутри класса. Например:

```
class MyClass {  
    private:  
        // Приватные члены  
  
    public:  
        // Публичные члены  
  
    protected:  
        // Защищенные члены  
};
```

Однако во многих кодстайл-гайдах и соглашениях о стиле написания кода принято следовать некоторому порядку, например, сначала объявлять публичные члены, затем приватные, и, если необходимо, защищенные.

Это может улучшить читаемость кода и облегчить его анализ.

Модификаторы в Swift

В Swift есть три основных модификатора доступа, аналогичных `public`, `private` и `protected` в C++:

- `public`: Элементы с этим модификатором доступа доступны из любого места вашего кода, а также из внешних модулей (пакетов).
- `internal`: Это модификатор доступа по умолчанию. Элементы с таким модификатором доступа доступны только в пределах текущего модуля.
- `private`: Элементы с этим модификатором доступа доступны только в пределах текущего файла.

`fileprivate`: Элементы с этим модификатором доступа доступны в пределах одного файла

Примеры использования модификаторов на Swift

```
class Person {  
    private var name: String  
    var age: Int  
  
    init(name: String, age: Int) {  
        self.name = name  
        self.age = age  
    }  
  
    func introduce() {  
        print("My name is \(name) and I am  
              \(age) years old.")  
    }  
  
    private func someFunc() {  
        print("")  
    }  
}
```