



Вау! ИИ готовит к ЕГЭ по информатике

[Попробовать](#)

Урок Спринт 1.9

Полиморфизм

Благодаря **полиморфизму** у объектов разных классов одинаковый интерфейс, но ведут они себя по-разному в зависимости от контекста.

В Go принцип полиморфизма заложен в интерфейсах. Это наборы методов, которые может реализовать любой тип. Если он реализует все методы интерфейса, этот тип можно использовать везде, где ожидается этот интерфейс:

```
package main

import "fmt"

type Animal interface {
    Speak() string
}

type Dog struct {
    Name string
}

func (d *Dog) Speak() string {
    return "Woof!"
}

type Cat struct {
    Name string
}

func (c *Cat) Speak() string {
    return "Meow!"
}

type Cow struct {
    Name string
}

func (c *Cow) Speak() string {
    return "Moo!"
}

func main() {
    animals := []Animal{
        &Dog{"Fido"},
        &Cat{"Fluffy"},
        &Cow{"Betsy"},
    }

    for _, animal := range animals {
        fmt.Println(animal.Speak())
    }
}
```

Мы определяем интерфейс `Animal` с методом `Speak()`. Затем — три типа, `Dog`, `Cat` и `Cow`, каждый из которых реализует метод `Speak()` интерфейса `Animal`. Далее мы создаём слайс со всеми тремя типами, проходим по нему в цикле и вызываем `Speak()` для каждого элемента. Результатом будут "Woof!", "Meow!" и "Moo!".

Перед нами пример полиморфизма: все три типа реализуют один и тот же интерфейс `Animal`, но каждый из них ведёт себя по-разному в зависимости

от типа. Это делает код гибче и более расширяемым, мы легко можем добавлять новые типы, которые реализуют интерфейс `Animal`, и использовать их вместо нынешних типов.

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение](#).

© 2018 – 2024 ООО «Яндекс»