



Ву! ИИ готовит к ЕГЭ по информатике

[Попробовать](#)

Урок Спринт 1.10

Кодогенерация

Кодогенерация — это генерация кода на основе шаблонов и данных.

Она пригодится в случаях, когда нужно создать много кода с одинаковой структурой и различиями в некоторых параметрах. С помощью кодогенерации вы можете создавать операции для базы данных или файлы конфигурации (например, для Kubernetes — системы разворачивания приложений в кластере).

В Go для кодогенерации можно использовать стандартный пакет `text/template`. Он позволяет создавать шаблоны, заполнять их данными и генерировать код.

Предположим, мы создаём CRUD-операции (Create-Read-Update-Delete, стандартный набор операций) для базы данных. У нас есть структура `User`:

```
type User struct {
    ID      int
    Name    string
    Email   string
    Password string
}
```

Мы хотим сгенерировать функции, которые будут создавать, читать, обновлять и удалять пользователей. Вот какой шаблон нам для этого понадобится:

```
package {{.Package}}

import (
    "database/sql"
    "errors"
)

type {{.Struct}} struct {
    db *sql.DB
}

func New{{.Struct}}(db *sql.DB) *{{.Struct}} {
    return &{{.Struct}}{db: db}
}

func (s *{{.Struct}}) Create{{.Struct}}(u *{{.User}}) error {
    // Создаёт пользователя в базе данных
}

func (s *{{.Struct}}) Read{{.Struct}}(id int) (*{{.User}}, error) {
    // Читает пользователя из базы данных
}

func (s *{{.Struct}}) Update{{.Struct}}(u *{{.User}}) error {
    // Обновляет пользователя в базе данных
}

func (s *{{.Struct}}) Delete{{.Struct}}(id int) error {
    // Удаляет пользователя из базы данных
}
```

Затем мы можем заполнить этот шаблон данными и сгенерировать код для нашей структуры `User`:

```
package main
```

```
import (
    "os"
    "text/template"
)

type User struct {
    ID      int
    Name    string
    Email   string
    Password string
}

func main() {
    t := template.Must(template.ParseFiles("template.go.tpl"))
    data := struct {
        Package string
        Struct  string
        User     string
    }{
        Package: "db",
        Struct:  "UserRepository",
        User:    "User",
    }
    t.Execute(os.Stdout, data)
}
```

Здесь мы используем пакет `text/template`, чтобы загрузить шаблон из файла `template.go.tpl`. Затем мы создаём данные, чтобы заполнить шаблон: имя пакета, имя структуры и имя нашей структуры `User`. А чтобы заполнить шаблон этими данными и вывести результат в консоль, мы добавляем метод `Execute`.

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»