

Харківський університет радіоелектроніки Факультет
комп'ютерних наук
Кафедра програмної інженерії ЗВІТ
до практичного заняття з дисципліни "Аналіз
та рефакторинг коду"
на тему: "Мова програмування Go. Методи рефакторингу коду "

Виконав ст. гр ПЗП-22-2
Єременко Андрій Віталійович

Перевірив доцент кафедри ПІ
Лещинський Володимир Олександрович

Харків 2024

МЕТА РОБОТИ

Мета – ознайомитися з основними методами рефакторингу коду в Go. Ми розглянемо три основні проблеми, які можуть виникати при написанні коду, і покажемо, як ці проблеми можна вирішити за допомогою простих та ефективних технік рефакторингу.

ВИСНОВКИ

Рефакторинг коду підвищує його якість, роблячи його простішим для розуміння, більш гнучким і легшим у підтримці, що сприяє покращенню ефективності розробки програмного забезпечення.

Додаток А

Слайди презентації

Refactoring Methods in Go

Підготував Єременко Андрій. 2024р

90%

“програмістів вважають рефакторинг критично важливим для довгострокового успіху проекту.” — Глобальне опитування розробників, 2023

Agenda

Розглянемо 3 метода рефакторингу:

- Self Encapsulate Field
- Rename Method
- Replace Temp with Query

Self Encapsulate Field

- Що це?

Метод рефакторингу, який полягає в тому, щоб заховати доступ до поля через методи (гетери та сетери), замість прямого доступу до поля ззовні.

Це дозволяє зберегти контроль над змінами значень полів і полегшує підтримку коду.

- Коли використовувати?
 - Якщо потрібно додати додаткову логіку при зміні значення поля (перевірка, логування тощо).
 - Якщо потрібно змінити внутрішнє представлення об'єкта без зміни зовнішнього інтерфейсу.

Приклад

```
1 package model
2
3 // Person is short information about human.
4 type Person struct { 3 usages new *
5     age int
6 }
7
8 func (p *Person) Age() int { new *
9     return p.age
10 }
11
12 func (p *Person) SetAge(age int) { no usages new *
13     if age >= 0 {
14         p.age = age
15     }
16 }
17
```

Rename Method

- Що це?

Метод рефакторингу, який полягає у зміні імені методу, щоб зробити його більш зрозумілим та відповідним до того, що він реально виконує.

Це покращує читабельність коду, особливо для нових розробників, що працюють з проектом.

- Коли використовувати
- Якщо ім'я методу не відповідає його функціоналу.
- Якщо ім'я методу не є достатньо зрозумілим для інших розробників.

Приклад

```
20 // ProcessData bad function name.  
21 func ProcessData(a int, b int) int { 2 usages new *  
22     return a + b  
23 }  
24  
25 // AddNumbers is a good function name.  
26 func AddNumbers(a int, b int) int { 2 usages new *  
27     return a + b  
28 }
```

Replace Temp with Query

- Що це?

Метод рефакторингу, який полягає в заміні тимчасових змінних на виклики методів або функцій, що безпосередньо виконують необхідну операцію. Це дозволяє зменшити кількість змінних у функціях та зробити код більш читабельним і зручним для підтримки.

- Коли використовувати?
- Якщо тимчасова змінна використовується тільки один раз.
- Якщо можна замінити змінну викликом методу або виразу, що робить код більш чітким.

Приклад

```
20 // Using of temporary variable.  
21 func calculateDiscountVariable(price float64) float64 { 1 usage new *  
22     discount := price * 0.1  
23     return price - discount  
24 }  
25  
26 // Using of a temporary query.  
27 func calculateDiscountQuery(price float64) float64 { 1 usage new *  
28     return price - (price * 0.1)  
29 }
```

Дякую за увагу.

Підготував Єременко Андрій. 2024р