

Министерство цифрового развития, связи и массовых коммуникаций  
Российской Федерации  
Ордена Трудового Красного Знамени федеральное государственное  
бюджетное образовательное учреждение высшего образования  
«Московский технический университет связи и информатики»  
(МТУСИ)

Кафедра «Сетевых Информационных Технологий и Сервисов»  
дисциплина «Рефакторинг баз данных и приложений»

Лабораторная работа № 1  
**«Рефакторинг приложений с  
использованием обратного  
проектирования»**

Выполнил: студент группы БПИ2303  
Берездовец Илья Сергеевич

**Цель работы:**

Ознакомиться с основными принципами рефакторинга и обратного проектирования. Научиться анализировать существующий код, выявлять его слабые стороны и предлагать улучшения. Применить на практике методы рефакторинга для улучшения читаемости, структуры и производительности кода. Развить навыки документирования изменений в коде.

**Ход работы:**

**Изначальный код:**

```

1 import UIKit
2
3 final class MovieQuizViewController: UIViewController {
4
5     @IBOutlet weak var yesButton: UIButton!
6     @IBOutlet weak var noButton: UIButton!
7
8     @IBOutlet weak var labelCount: UILabel!
9     @IBOutlet weak var imageView: UIImageView!
10    @IBOutlet weak var textLabel: UILabel!
11
12    var currentQuestionIndex = 0
13    var correctAnswers = 0
14
15    let textQuiz = "Рейтинг этого фильма больше чем 6?"
16
17    struct QuizQuestion {
18        let image: String
19        let text: String
20        let correctAnswer: Bool
21    }
22
23    let questions = [
24        QuizQuestion(image: "The Godfather", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: true),
25        QuizQuestion(image: "The Dark Knight", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: true),
26        QuizQuestion(image: "Kill Bill", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: true),
27        QuizQuestion(image: "The Avengers", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: true),
28        QuizQuestion(image: "Deadpool", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: true),
29        QuizQuestion(image: "The Green Knight", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: true),
30        QuizQuestion(image: "Old", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: false),
31        QuizQuestion(image: "The Ice Age Adventures of Buck Wild", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: false),
32        QuizQuestion(image: "Tesla", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: false),
33        QuizQuestion(image: "Vivarium", text: "Рейтинг этого фильма больше чем 6?", correctAnswer: false)
34    ]
35
36    override func viewDidLoad() {
37        super.viewDidLoad()
38        showQuestion()
39    }
40
41    func showQuestion() {
42        let question = questions[currentQuestionIndex]
43        imageView.image = UIImage(named: question.image) ?? UIImage()
44        textLabel.text = question.text
45        labelCount.text = "\(currentQuestionIndex + 1) / \(questions.count)"
46        imageView.layer.borderWidth = 0
47        yesButton.isEnabled = true
48        noButton.isEnabled = true
49    }
50
51    func showResult() {
52        let alert = UIAlertController(title: "Игра окончена", message: "Ваш результат: \(correctAnswers) из \(questions.count)", preferredStyle: .alert)
53        let action = UIAlertAction(title: "Сыграть снова", style: .default) { _ in
54            self.currentQuestionIndex = 0
55            self.correctAnswers = 0
56            self.showQuestion()
57        }
58    }
59

```

```

57     }
58     alert.addAction(action)
59     present(alert, animated: true)
60 }
61
62 func checkAnswer(isCorrect: Bool) {
63     yesButton.isEnabled = false
64     noButton.isEnabled = false
65
66     if isCorrect {
67         correctAnswers += 1
68     }
69
70     imageView.layer.borderWidth = 8
71     imageView.layer.borderColor = isCorrect ? UIColor.green.cgColor : UIColor.red.cgColor
72
73     DispatchQueue.main.asyncAfter(deadline: .now() + 1.0) {
74         if self.currentQuestionIndex == self.questions.count - 1 {
75             self.showResult()
76         } else {
77             self.currentQuestionIndex += 1
78             self.showQuestion()
79         }
80     }
81 }
82
83 @IBAction func yesButtonClicked(_ sender: Any) {
84     checkAnswer(isCorrect: questions[currentQuestionIndex].correctAnswer == true)
85 }
86
87 @IBAction func noButtonClicked(_ sender: Any) {
88     checkAnswer(isCorrect: questions[currentQuestionIndex].correctAnswer == false)
89 }
90 }

```

**Код после рефакторинга:**

```

1 import UIKit
2
3 final class MovieQuizViewController: UIViewController {
4     // MARK: - Lifecycle
5
6
7     @IBOutlet weak var yesButton: UIButton!
8     @IBOutlet weak var noButton: UIButton!
9     @IBOutlet weak var labelCount: UILabel!
10    @IBOutlet weak var imageView: UIImageView!
11    @IBOutlet weak var textLabel: UILabel!
12    override func viewDidLoad() {
13        super.viewDidLoad()
14        let firstQuestion = questions[currentQuestionIndex]
15        let viewModel = convert(model: firstQuestion)
16        show(quiz: viewModel)
17    }
18
19    private var currentQuestionIndex = 0
20    private var correctAnswers = 0
21    private static let textQuiz = "Рейтинг этого фильма больше чем 6?"
22
23    struct QuizResultsViewModel{
24        let title: String
25        let text: String
26        let buttonText: String
27    }
28
29    struct QuizQuestion{
30        let image: String
31        let text: String
32        let correctAnswer: Bool
33    }
34
35    struct QuizStepViewModel {
36        let image: UIImage
37        let question: String
38        let questionNumber: String
39    }
40
41    private let questions: [QuizQuestion] = [
42        QuizQuestion(
43            image: "The Godfather",
44            text: textQuiz,
45            correctAnswer: true),
46        QuizQuestion(
47            image: "The Dark Knight",
48            text: textQuiz,
49            correctAnswer: true),
50        QuizQuestion(
51            image: "Kill Bill",
52            text: textQuiz,
53            correctAnswer: true),
54        QuizQuestion(
55            image: "The Avengers",
56            text: textQuiz,
57            correctAnswer: true),
58        QuizQuestion(
59            image: "Deadpool",
60            text: textQuiz,
61            correctAnswer: true),
62        QuizQuestion(
63            image: "The Green Knight",
64            text: textQuiz,
65            correctAnswer: true),
66        QuizQuestion(

```

```

3  final class MovieQuizViewController: UIViewController {
41  private let questions: [QuizQuestion] = [
66      QuizQuestion(
67          image: "Old",
68          text: textQuiz,
69          correctAnswer: false),
70      QuizQuestion(
71          image: "The Ice Age Adventures of Buck Wild",
72          text: textQuiz,
73          correctAnswer: false),
74      QuizQuestion(
75          image: "Tesla",
76          text: textQuiz,
77          correctAnswer: false),
78      QuizQuestion(
79          image: "Vivarium",
80          text: textQuiz,
81          correctAnswer: false)
82  ]
83
84  private func show(quiz step: QuizStepViewModel){
85      imageView.image = step.image
86      labelCount.text = step.questionNumber
87      textLabel.text = step.question
88
89      imageView.layer.borderWidth = 0
90
91      yesButton.isEnabled = true
92      noButton.isEnabled = true
93  }
94
95  private func show(quiz result: QuizResultsViewModel) {
96      let alert = UIAlertController(
97          title: result.title,
98          message: result.text,
99          preferredStyle: .alert)
100
101      let action = UIAlertAction(title: result.buttonText, style: .default) { _ in
102          self.currentQuestionIndex = 0
103          self.correctAnswers = 0
104
105          let firstQuestion = self.questions[self.currentQuestionIndex]
106          let viewModel = self.convert(model: firstQuestion)
107          self.show(quiz: viewModel)
108      }
109
110      alert.addAction(action)
111
112      self.present(alert, animated: true, completion: nil)
113  }
114
115  private func convert(model: QuizQuestion) -> QuizStepViewModel{
116      let questionStep = QuizStepViewModel(
117          image: UIImage(named: model.image) ?? UIImage(),
118          question: model.text,
119          questionNumber: "\(currentQuestionIndex + 1)/\(questions.count)"
120      )
121      return questionStep
122  }
123
124  private func showNextQuestionOrResults(){
125
126      if currentQuestionIndex == questions.count - 1 {
127
128          let alert = UIAlertController(
129              title: "Этот раунд окончен!",
130              message: "Результат: \(correctAnswers)/\(questions.count)"

```

```

3 final class MovieQuizViewController: UIViewController {
124 private func showNextQuestionOrResults(){
126     if currentQuestionIndex == questions.count - 1 {
127
128         let alert = UIAlertController(
129             title: "Зрор пауна окончен!",
130             message: "Ваа паулыат \((correctAnswers)/\((questions.count))",
131             preferredStyle: .alert)
132
133         let action = UIAlertAction(title: "Сырпать еше паа", style: .default) { _ in
134             self.currentQuestionIndex = 0
135             self.correctAnswers = 0
136             let firstQuestion = self.questions[self.currentQuestionIndex]
137             let viewModel = self.convert(model: firstQuestion)
138             self.show(quiz: viewModel)
139         }
140
141         alert.addAction(action)
142         self.present(alert, animated: true, completion: nil)
143
144     } else {
145         currentQuestionIndex += 1
146         let nextQuestion = questions[currentQuestionIndex]
147         let viewModel = convert(model: nextQuestion)
148         show(quiz: viewModel)
149     }
150 }
151
152 private func showAnswerResult(isCorrect: Bool) {
154     yesButton.isEnabled = false
155     noButton.isEnabled = false
156
157     if isCorrect{
158         correctAnswers += 1
159     }
160
161     imageView.layer.masksToBounds = true
162     imageView.layer.borderWidth = 8
163     imageView.layer.borderColor = isCorrect ? UIColor.ygreen.cgColor : UIColor.yred.cgColor
164
165     DispatchQueue.main.asyncAfter(deadline: .now() + 1.0) {
166         self.showNextQuestionOrResults()
167     }
168 }
169
170 }
171
172 @IBAction func yesButtonClicked(_ sender: Any) {
173     let currentQuestion = questions[currentQuestionIndex]
174     let givenAnswer = true
175     showAnswerResult(isCorrect: givenAnswer == currentQuestion.correctAnswer)
176 }
177
178 @IBAction func noButtonClicked(_ sender: Any) {
179     let currentQuestion = questions[currentQuestionIndex]
180     let givenAnswer = false
181     showAnswerResult(isCorrect: givenAnswer == currentQuestion.correctAnswer)
182 }
183
184 }
185 /*
186 Моск-данные
187
188
189 Картинка: The Godfather

```

## Улучшения после рефакторинга:

- 1) Код стал более читаем и структурирован.
- 2) Все переменные и функции помечены модификатором `private`. Тем самым я сделал код безопаснее от случайных изменений переменных и методов вне класса.
- 3) Создал переменную `textQuiz` строкового типа куда положил текст вопроса(так как вопрос везде одинаков).
- 4) Написал Структуры `QuizStepViewModel` и `QuizResultsViewModel` для разделения логики показа экрана вопроса и результата игры (квиза). Это дает гибкость в показе данных. Каждая структура не зависима друг от друга.
- 5) Добавлены методы `show`, которые представляют показ данных. Это удобно тем, что происходит отделение данных от отображения, которые будут удобны для UI.

6) Добавлена функция `convert`, которая конвертирует данные с `QuizQuestion` и показывает их при переходе на следующий вопрос с новыми данными исходя из `QuizQuestion`.

Диаграмма логики работы приложения.



Ссылка на GitHub: <https://github.com/iberezaa/MovieQuiz>