|  |  |
| --- | --- |
| **Символика** | **Логотип ПИКС_3** |

**ВОПРОСЫ К ЗАЧЕТУ**

**по дисциплине**

**«РЕФАКТОРИНГ И ОПТИМИЗАЦИЯ КОДА»**

**Весенний семестр 2024-2025 учебного года**

**Специальность 1-40 05 01-10 Информационные системы и технологии  
(в бизнес-менеджменте)**

**(группа 214371)**

1. Определение рефакторинга. Особенности проектирования программных продуктов.
2. Модели жизненного цикла программных продуктов.
3. Рефакторинг и проектирование программных продуктов.
4. Рефакторинг и производительность.
5. Условия, требующие применение рефакторинга.
6. «Правило трех» в рефакторинге.
7. Рефакторинг при добавлении функции.
8. Рефакторинг при исправлении ошибок программного кода.
9. Проблемы при выполнении рефакторинга.
10. Изменения проекта, затрудняющие рефакторинг.
11. Условия, исключающий применение рефакторинга.
12. Дублируемый код. Длинный метод.
13. Большой класс. Длинный список параметров.
14. Расходящиеся изменения. Стрельба дробью.
15. Завистливые функции. Группы данных.
16. Одержимость примитивами. Инструкции switch.
17. Параллельные иерархии наследования. «Ленивый класс».
18. Теоретическая общность. Временное поле.
19. Цепочки сообщений.
20. Альтернативные классы с разными интерфейсами.
21. Неполный библиотечный класс.
22. Классы данных. Отказ от наследства.
23. Важность самотестируемого кода.
24. Каркас тестирования Junit.
25. Модульные и функциональные тесты. Добавление новых тестов.
26. Формат описания рефакторинга.
27. Поиск ссылок. Зрелость методов рефакторинга.
28. Извлечение метода. Встраивание метода.
29. Встраивание временной переменной. Замена временной переменной запросом.
30. Введение поясняющей переменной. Расщепление временной переменной.
31. Удаление присваиваний параметрам. Замена метода объектом методов.
32. Замена алгоритма.
33. Перенос метода. Перенос поля.
34. Извлечение класса. Встраивание класса.
35. Сокрытие делегирования.
36. Удаление посредника.
37. Введение внешнего метода. Введение локального расширения.
38. Самоинкапсуляция поля.
39. Замена значения данных объектом. Замена значения ссылкой. Замена ссылки значением.
40. Замена массива объектом.
41. Дублирование видимых данных.
42. Замена однонаправленной связи двунаправленной.
43. Замена двунаправленной связи однонаправленной. Замена магического числа символьной константой.
44. Инкапсуляция поля. Инкапсуляция коллекции.
45. Перемещение поведения в класс. Замена записи классом данных.
46. Замена кода типа классом. Замена кода типа подклассами.
47. Замена кода типа состоянием/стратегией.
48. Замена подкласса полями.
49. Декомпозиция условного оператора.
50. Консолидация условного выражения. Консолидация дублирующихся условных фрагментов.
51. Удаление управляющего флага.
52. Замена вложенных условных операторов граничным оператором.
53. Замена условной инструкции полиморфизмом.
54. Введение нулевого объекта. Введение утверждения.
55. Переименование метода.
56. Добавление параметра. Удаление параметра.
57. Разделение запроса и модификатора.
58. Вопросы многопоточности.
59. Параметризация метода. Замена параметра явными методами.
60. Сохранение всего объекта. Замена параметра вызовом метода.
61. Введение объекта параметра. Удаление метода установки значения.
62. Сокрытие метода.
63. Замена конструктора фабричным методом.
64. Инкапсуляция нисходящего приведения типа.
65. Замена кода ошибки исключением. Замена исключения проверкой.
66. Подъем поля. Подъем метода. Подъем тела конструктора.
67. Опускание метода. Опускание поля.
68. Извлечение подкласса. Извлечение суперкласса.
69. Извлечение интерфейса.
70. Свертывание иерархии.
71. Формирование шаблонного метода.
72. Замена наследования делегированием. Замена делегирования наследованием.
73. Важность крупномасштабных рефакторингов.
74. Четыре крупномасштабных рефакторинга.
75. Разделение наследования. Преобразование процедурного проекта в объектный.
76. Отделение предметной области от представления. Выделение иерархии.
77. Проверка в реальных условиях.
78. Как и когда применять рефакторинг? Рефакторинг как средство получения скорейших выгод.
79. Уменьшение стоимости рефакторинга.
80. Безопасный рефакторинг. Проверка в реальных условиях.
81. Ресурсы и ссылки, относящиеся к рефакторингу.
82. Следствия повторного использования программного обеспечения и передачи технологий.
83. Рефакторинг с помощью инструментов.
84. Технические критерии инструментария для рефакторинга.
85. Деревья синтаксического анализа.
86. Практические критерии инструментария. Отмена модификаций. Интеграция с другими инструментами.

**ЛИТЕРАТУРА**

1. Босуэлл, Д. Фаучер, Т. Читаемый код, или Программирование как искусство / Д. Босуэлл, Т. Фаучер. – СПб. и др.: Питер, 2012. – 203 с.
2. Макконнелл, С. Совершенный код : практическое руководство по разработке программного обеспечения / С. Макконнелл. – Санкт-Петербург : Питер ; Москва : Русская редакция, 2007. – 896 с. : ил.
3. Мартин, Р. Чистый код: создание, анализ и рефакторинг / Р. Мартин. – Санкт-Петербург : Питер, 2024. – 464 с.
4. Орлов, С.А. Программная инженерия : учебник / С. А. Орлов. – 5-е изд., обновл. и доп. – Санкт-Петербург : Питер, 2017. – 640 с.
5. От хранения данных к управлению информацией : учебник / пер. с англ. Н. Вильчинского. – 2-е изд. – Санкт-Петербург : Питер, 2016. – 544 с.
6. Фаулер, М. Рефакторинг кода на JavaScript: улучшение проекта существующего кода / М. Фаулер, 2-е изд.: пер. с англ. – СПб.: ООО «Диалектика», 2019. – 464 с.
7. Фаулер, М. Рефакторинг: улучшение существующего кода. –  
   пер. с англ. – СПб: Символ Плюс, 2003. – 432 с.

Вопросы разработала:

ТОНКОВИЧ Ирина Николаевна – канд.хим.наук, доцент