|  |  |
| --- | --- |
| **E:\!Кафедра ПИКС\Логотип БГУИР\Символика.jpg** | **E:\!Кафедра ПИКС\Логотип ПИКС\17 мая 2013\Логотип ПИКС_3.jpg** |

**ВОПРОСЫ К ЗАЧЕТУ**

**по дисциплине**

**«оБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ  
И ПРОГРАММИРОВАНИЕ»**

**Весенний семестр 2023-2024 учебного года**

**Специальность 6-05-0611-01 01 Информационные системы и технологии**

1. Философия ООП. Определения ООП. Отличительные характеристики объектно-ориентированной программы.
2. Основополагающие принципы ООП. Инкапсуляция.
3. Основополагающие принципы ООП. Наследование. Управление наследованием.
4. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
5. Абстракция в ООП.
6. Раннее и позднее связывание.
7. Переменные и базовые типы данных. Примитивные и ссылочные типы.
8. Классы-оболочки.
9. Приведение типов (явное и автоматическое). Константы и переменные.
10. Операторы в Java. Порядок действий (приоритет операторов).
11. Операторы ветвления. Циклы.
12. Массивы. Способы объявления и инициализации массивов. Индексация и размер массива.
13. Встроенный класс String. Строковые операции.
14. Классы StringBuilder и StringBuffer.
15. Методы для работы со строками.
16. Регулярные выражения.
17. Перечисления.
18. Классы и объекты. Абстракция данных.
19. Переменные класса и константы. Ограничение доступа.
20. Модификаторы уровня доступа.
21. Специальные методы классов (конструкторы). Конструктор по умолчанию. Конструкторы при наследовании в Java.
22. Класс Object. Основные методы класса Object
23. Ключевое слово static. Ключевое слово final.
24. Методы. Статические методы и атрибуты.
25. Передача объектов в методы. Перегрузка и переопределение методов.
26. Использование super и this.
27. Абстрактные классы и методы.
28. Внутренние (inner) классы.
29. Вложенные (nested) классы.
30. Анонимные (anonymous) классы.
31. Интерфейсы.
32. Пакеты. Логические блоки.
33. Records.
34. Иерархия и способы обработки исключительных ситуаций.
35. Перехват и обработка исключения в блоке try-catch. Вложенные операторы try.
36. Оператор throw. Ключевое слово finally.
37. Оператор throws. Собственные исключения.
38. Общее представление о потоках ввода/вывода.
39. Байтовые потоки ввода.
40. Байтовые потоки вывода.
41. Символьные потоки ввода.
42. Символьные потоки вывода.
43. Классы PrintStream и PrintWriter.
44. Сериализация и десериализация.
45. Класс File.
46. Класс Scanner.
47. Коллекции. Общая характеристика. Алгоритмы класса Collections.
48. Списки. Иерархия классов.
49. Множества. Иерархия классов.
50. Карты отображений. Иерархия классов.
51. Универсальные классы и методы.
52. Универсальные интерфейсы. Наследование и универсальные типы.
53. Интерфейс Comparable. Интерфейс Comparator.
54. Интерфейс Iterator. Интерфейс ListIterator.
55. Лямбда-выражения.
56. Функциональный интерфейс. Передача метода в качестве параметра.
57. Функциональные интерфейсы: Predicate, Function, BinaryOperator.
58. Функциональные интерфейсы: Consumer, Supplier, UnaryOperator.
59. Понятие многопоточности.
60. Создание и выполнение потоков в Java.
61. Класс Thread.
62. Интерфейс Runnable.
63. Синхронизация потоков. Оператор synchronized.
64. Синхронизация потоков. Класс Phaser.
65. Синхронизация потоков. Блокировки. ReentrantLock
66. Методы wait(), notify(), notifyAll().
67. Семафоры.
68. Обмен между потоками. Класс Exchanger.
69. Основы Stream API.
70. Stream API. Создание потока данных
71. Stream API. Метод collect.
72. Stream API. Тип Optional.
73. Stream API. Получение подпотока и объединение потоков.
74. Понятие паттерна проектирования.Классификация паттернов.
75. Паттерн «Делегирование».
76. Паттерн «Функциональный дизайн».
77. Паттерн «Неизменяемый интерфейс».
78. Паттерн «Интерфейс».
79. Паттерн «Интерфейс-маркер».
80. Паттерн «Контейнер свойств».
81. Паттерн «Канал событий».
82. Архитектурный шаблон MVC.
83. Архитектурный шаблон MVP.
84. Архитектурный шаблон MVVM.
85. Архитектурный шаблон HMVC.
86. Порождающий паттерн «Фабричный метод».
87. Порождающий паттерн «Абстрактная фабрика».
88. Порождающий паттерн «Строитель».
89. Порождающий паттерн «Прототип».
90. Порождающий паттерн «Одиночка».
91. Структурный паттерн «Адаптер».
92. Структурный паттерн «Мост».
93. Структурный паттерн «Компоновщик».
94. Структурный паттерн «Декоратор».
95. Структурный паттерн «Фасад».
96. Поведенческий паттерн «Цепочка обязанностей».
97. Поведенческий паттерн «Итератор».
98. Поведенческий паттерн «Снимок».
99. Поведенческий паттерн «Состояние».
100. Поведенческий паттерн «Шаблонный метод».
101. Поведенческий паттерн «Команда».
102. Поведенческий паттерн «Посредник».
103. Поведенческий паттерн «Наблюдатель».
104. Поведенческий паттерн «Стратегия».
105. Поведенческий паттерн «Посетитель».
106. Принципы SOLID.

Разработал:

МИГАЛЕВИЧ Сергей Александрович –

магистр технических наук, старший преподаватель