**Вопросы к экзамену по курсу**

**«Объектно-ориентированные технологии программирования и стандарты проектирования ч.2»**

1. Особенности CASE-технологии. История развития CASE-средств.
2. Системная модель CASE-средств.
3. Критерии развития CASE-средств.
4. Понятие проекта. Масштаб проекта. Общие принципы управления проектом.
5. Три составляющие программного проекта: система обозначений, процесс и инструмент. Их роль и значение для проекта.
6. Типы и особенности современных программных проектов.
7. Задачи и категории современных методологий создания программных проектов.
8. Взаимосвязь между методологией, размером задачи и командой разработчиков.
9. Целесообразность использования различных методологий для различных типов программных проектов.
10. Унифицированный процесс разработки программных средств.
11. Основные и дополнительные элементы объектно-ориентированного подхода.
12. История появления, особенности и назначение унифицированного языка моделирования UML.
13. Требования к программному обеспечению. Первичные и детальные требования. Функциональные и нефункциональные требования.
14. Назначение, особенности и построение диаграммы *Use Case*.
15. Назначение, особенности и построение диаграммы *Deployment*.
16. Назначение, особенности и построение диаграммы *Statechart*.
17. Назначение, особенности и построение диаграммы *Activity*.
18. Назначение, особенности и построение диаграммы *Sequence*.
19. Назначение, особенности и построение диаграммы *Collaboration*.
20. Назначение, особенности и построение диаграммы *Component*.
21. Назначение, особенности и построение диаграммы *Class*, виды и особенности связей между классами на диаграммах.
22. Понятие шаблонов проектирования и их классификация. Шаблоны в нотации языка UML.
23. Шаблон “Фасад” и его обозначение в нотации языка UML.
24. Шаблон “Наблюдатель” и его обозначение в нотации языка UML.
25. Декомпозиция программной системы на модули. Принцип модульности. Оценка сложности программной системы через принцип модульности. Затраты на работу с модулями.
26. Определение модуля. Связность и сцепление модулей. Типы связности и сцепления модулей.
27. Создание модели предметной области программной системы с помощью диаграммы классов.
28. Создание модели анализа с помощью диаграммы классов. Различные стереотипы для классов и их назначение.
29. Декомпозиция системы. Правила выделения подсистем.
30. Особенности создания шаблона приложения в среде *Rational Rose* с использованием библиотеки *MFC*. Структура и классы приложения.
31. Функциональные возможности *Rational Rose*: модуль *Component Assignment Too*l, компонент *Model Assistant*, обновление кода по модели и модели по коду.
32. Особенности генерации исходного кода в среде *Rational XDE*. Способы синхронизации модели.
33. Назначение, возможности, особенности использования модуля *Web Modeler* в *Rational Rose*.
34. Возможности и особенности построения *Web*-модели в среде *Rational XDE*.
35. Понятие проектных рисков. Действия по управлению рисками.
36. Статический и динамический аспекты *Rational Unified Process (RUP)*.
37. Принципы и стадии разработки ПС в технологии *Rational Unified Process.*
38. Содержание и результаты первой и второй стадий в технологии *Rational Unified Process.*
39. Содержание и результаты третей и четвертой стадий в технологии *Rational Unified Process*.
40. Этапы создания программных средств в технологии *Oracle*.
41. Процессы создания программных средств в технологии *Oracle.*
42. Сравнительный анализ технологий создания ПС *Rational Unified Process*, *Oracle*, *Borland*.
43. Понятия CASE-средство, CASE-система, CASE-технология, CASE-индустрия и различия между ними.