



Внедрение в проектный путь дизайна, ориентированного на объект, не так просто, и для повторного использования его становится еще труднее.

Необходимо выбирать подходящие объекты, относить их к различным классам, сохранять их разумную степень детализации. Проектирование должно, с одной стороны, быть совместимым с задачей, которую необходимо выполнить, и быть общим для удовлетворения всех требований, которые могут возникнуть в

будущем. Однако проектировщики могут создать хорошую систему проектирования, хотя они шокированы тем, что могут предложить ряд вариантов. Сценариям нелегко придумать совершенно новую историю, вместо этого они берут за основу шаблоны, жанры и изображения, которые были разработаны в мировой литературе. Мотивом убийства являются деньги, месть, ревность и т.д. Это также в объектно-ориентированном дизайне, это жалко. Из этой книги вы не знаете, как создавать пользовательские интерфейсы, как писать устройства и как работать с базами данных, ориентированными на объект. Кроме того, отсутствует информация о жалких людях, характерных для конкретных тематических областей. В каждой из этих областей могут существовать их собственные жалкие устройства, и, возможно, кто-то их систематизировал. В дизайне, ориентированном на объект, повторное использование часто является важным фактором. Результаты являются результатом применения жалкого и различного рода компромиссов. Перерасчет всех эффектов поможет вам понять и оценить их роль. Что воспринимается от языка к строительному блоку. Очень важно выбрать язык программирования. В наших патологиях он будет использовать возможности Smalltalk и C++. В языке CLOS есть несколько методов, которые делают невозможной модель посетителя ненужной. MVC поддерживает встроенный вид с помощью класса ComputerView, который является подклассом класса Вид. Еще одна характеристика MVC заключается в том, что этот вид может быть включен. Например, допускается представление контрольной панели, состоящей из кнопок, в качестве композитного обзора, содержащего шаги, с помощью одной кнопки на каждой из них. ICC использует другие паттерны дизайна, такие как производственный метод для установления класса контрольного устройства по умолчанию для просмотра, и декоратор для добавления к виду вращаемости. Базовая связь в схеме ICC описывается паттерном в

наблюдателях, компостере и стратегии. Стратегия является объектом для представления алгоритма. Она полезна, когда вы хотите статическую или динамическую замену одного алгоритма на другой или когда алгоритм связан со сложными структурами данных. В приложениях содержится информация, которая поможет вам лучше понять путь и связанные с этим вопросы. Пример кода фрагментации, который иллюстрирует вероятность осуществления на языках C++ или Smalltalk. Результаты приложения, компромиссы, которые должны быть сделаны. Какие аспекты поведения системы могут быть независимо изменены с помощью этого типа? Рекомендации и рекомендуемые методы. Обычная (команда) инкапсулирует запрос как объект. Группирует объекты в структуры, похожие на деревья, чтобы представлять иерархический тип "serv-единица" позволяет клиентам работать с отдельными объектами в том же состоянии, что и набор объектов. Декораторы используются для расширения существующей функциональности. Государство (состояние) (291) допускает изменение поведения объекта при изменении его внутреннего состояния. "Страттаги" (Стратегия) (300) определяет положение семейства алгоритмов, инкапсулирует все из них и позволяет заменить одно из них другим. "Посетитель" (314) представляет операцию, которая должна быть проведена над элементами объекта. Только те, кто концентрируется только на межклассовых отношениях, считаются в категории "классовых патометров", отмечают, что большинство пасторов работают на уровне объектов. Поведение классов использует наследство для описания алгоритмов и потока управления. Поведение объектов описывает объекты как объекты, принадлежащие к определенной группе. Наиболее трудная задача в проектировании, ориентированном на объект, заключается в том, чтобы поместить систему в объекты. Интерфейс описывает все запросы, которые могут быть отправлены на объект. Тип - это название,

используемое для указания конкретного интерфейса. Другие модели указывают способ разложения объекта на более мелкие подобъекты. Абстрактный субобъект и строитель описывают объекты, единственная цель которых - создать другие объекты. Посетитель и команда - это объекты, ответственные за выполнение запроса на другой объект или группу. Факад модели показывает, как представить целую подсистему в качестве объекта. Адаптер модели показывает, как поддерживать большое число объектов с высокой степенью детализации. Динамическая связывание означает, что отправка определенной просьбы не определяет какой-либо конкретной реализации до момента выполнения. Различные объекты, поддерживающие один и тот же интерфейс, могут реагировать на такие запросы различными операциями. Такая эксплуатационная совместимость объектов называется полиморфизмом и является важной особенностью объектно-ориентированных систем. Она позволяет клиенту не делать практически никаких предположений в отношении объектов, за исключением того, что они поддерживают определенный объект, что сам полиморфизм может быть определен объектом и что он может быть принят решение о его выполнении в том же состоянии. Исполнение объекта определяется его классом. Класс указывает внутренние данные объекта и его представление. В нашей лекции, основанной на DMT (см. приложение B), класс может быть представлен в виде прямоугольника, в котором название класса написано жирным шрифтом. Операции перечислены ниже. Любые данные, определенные для класса, следуют после операций. Эта диаграмма может представлять собой псевдокод, описывающий выполнение операции. В таком случае код представляется в прямоугольнике со сложным углом, соединенным пунктирной линией с операцией, которую он выполняет. Класс объекта называется классом (класс миксина), цель которого состоит в обеспечении

дополнительного интерфейса или функциональности для других классов.

Название абстрактных классов курсивом отличается от конкретных. Заголовок также используется для указания абстрактных операций. В Smalltalk наследование понимается только как последовательность реализации.

Программное обеспечение C++ манипулирует объектами по типам, определяемым абстрактными классами. Истинное наследование интерфейса может быть применено к C++ с помощью открытого наследства абстрактному классу. В Smalltalk объектов, построенных в соответствии с жалкой цепью обязанностей, должен иметь общий тип, но их реализация, как правило, является иной. Существует два преимущества манипулирования объектами строго через абстрактный интерфейс класса. Клиент не нуждается в информации о конкретных типах объектов, которые он использует, при условии, что все они имеют ожидаемый интерфейс. Эти преимущества значительно сокращают количество связей между подсистемами. Возможно даже сформулировать принцип объектно-ориентированного дизайна для повторного использования. Не объявляйте переменные в этом месте в качестве копий конкретных классов. Вместо этого держитесь интерфейса, определенного классом. Его легче использовать, поскольку он непосредственно поддерживается языком программирования. В случае чередования классов также упрощается задача изменения существующего осуществления. Если подкласс заменяет только некоторые операции, то могут быть затронуты другие унаследованные операции, поскольку они могут вызвать замену. Однако подкласс и материнский класс настолько тесно связаны между собой, что любое изменение в последнем потребует изменения в осуществлении подкласса. По сравнению с составом объектов, а не последовательностью классов, вы включаете в себя каждый класс и позволяете ему выполнять только свою задачу. Классы и их иерархия остаются

небольшими, и вероятность их роста до неуправляемого размера невелика. Конструкция, основанная на составе, будет содержать больше объектов (хотя число классов может уменьшиться), и поведение системы будет зависеть от их взаимодействия. Окно не является прямоугольником, но содержит его. Теперь класс Windows может четко перенаправить запросы на свой член Rectangle, а не наследовать его операции. Внутренняя площадь окна может быть круглой во время реализации, просто путем предоставления вместо этого копии класса Rectangle. Предполагается, что оба класса имеют один и тот же тип конструкции. Делегация также является отличительной особенностью места и других подходов, применяемых для повышения гибкости в отношении состава объектов. Конструкция не одинакова, один и тот же тип дизайна. Один и тот же и тот же и другой. Другие подходы, которые должны применяться для разработки и другого подхода, чтобы сделать его более гибким и сделать его более гибким. И сделать его более гибким. При наличии существенного сходства между абстракцией и конкретным осуществлением абстракция может просто делегировать операции. Наследование и параметризованные типы являются другим (хотя и не совсем объективным) методом повторного использования имеющихся функциональных возможностей. Этот метод позволяет идентифицировать тип без указания типов, которые он использует. Неконкретные типы элементов представлены в перечне характеристик, имеющихся в опционах. Структуры времени и времени программы, ориентированной на объект, на стадии осуществления зачастую имеют мало общего со структурой ее исходного кода. Последний установлен на этапе компиляции; код состоит из классов, взаимоотношения которых не являются постоянными. Эти две структуры практически независимы. Можно обеспечить выполнение по умолчанию, которое может быть заменено подклассами. Однако ни наследование, ни параметризованные типы не могут

быть изменены на языке, таком, как Smalltalk. Различие между знаниями и агрегированием, скорее, зависит от предполагаемого использования, а не от языковых механизмов. Таким образом, в структуре этапа компиляции не так просто увидеть разницу, но она тем не менее значительна. Взаимосвязь между объектами и их типами должна быть очень тщательной, поскольку именно введение в их конструкцию зависит от степени успеха дизайна и провала структуры на этапе осуществления. Поведение системы на момент внедрения должно определяться дизайнером, а не языком. Время осуществления является трудным. Реконструирование системы может быть трудным, и дизайн объекта может быть трудным. Вот некоторые из типичных причин для перепроектирования, а также патологии, которые избегают этого. При создании объекта явно существует класс. При присвоении имени класса вы связываете себя с конкретной реализацией, а не с интерфейсом. Это может затруднить изменение объекта в будущем. Чтобы избежать такой проблемы, создайте объекты косвенно. Устанавливая конкретную операцию, вы ограничиваете себя единственным способом выполнения запроса. Если вы не включаете запросы в код, то будет легче изменить способ реагирования на запрос как на этапе компиляции, так и на этапе выполнения. Создание подклассов приводит к общему увеличению числа классов. Для расширения внутреннего повторного использования будут использоваться картеры, которые уменьшают количество взаимоотношений. Более низкая степень подключения также повышает способность системы развиваться. Расширение класса легче, если оно не зависит от многих других. Часто оно включает классы из одной или нескольких библиотек заранее определенных классов. Такие библиотеки называются инструментами. В рамках этой программы разрабатываются общие для данной тематической области решения по проектированию, основное внимание в рамках которой

уделяется повторному использованию дизайна, хотя и включает в себя конкретные подклассы, которые могут применяться непосредственно. Повторное использование на этом уровне изменяет взаимосвязь между приложением и его программным обеспечением. Третье позволит упростить разработку приложений для финансового моделирования. Рамки, в которых они решаются с помощью применения патана, могут лучше обеспечить высокий уровень дизайна и повторного использования кода. В использованных каркатурах можно найти несколько различных картеров дизайна. Структуры помогают адаптировать архитектуру солёности каркаупатов к различным видам применения.

Дополнительное преимущество заключается в том, что вместе с рамками эти патоны, которые в них используются, документально зафиксированы [B]94]. Системы, ориентированные на объект, могут быть в максимально возможной степени повторно использованы. Крупные прикладные программы, ориентированные на объект, получают из слоев, которые взаимодействуют друг с другом. Конструкция и код приложения в значительной степени получены от машин, применявшихся при его создании. Если дизайнер получает каталог из более чем 20 пасторов, то трудно решить, какой тип наиболее подходит для конкретной проектировочной задачи. В таблице 1.2 перечислены те аспекты конструкции, которые различаются от одного жалкого к другому. Этот подход не способствует изучению причин, приведших к необходимости реорганизации. Вместо этого, подумайте о том, что может изменить дизайн и что вы хотели бы изменить без перепроектирования. Все обычные рекомендации используются. Со временем вы будете разрабатывать свой собственный подход к работе с пейзажем. Невозможно будет использовать общий метод дизайна заводского дизайнера Прототипа сайта. Способность Аптерны, вероятно, изменит способность создавать путь. Единственный класс, из которого установлен объект.

Средства создания единого объекта подкласса конструкционного объекта.

Возможно изменить поведение пути поведения как путь к посадке, путь пути или путь, который вовсе не является дорожкой. Вы найдете совет о том, как лучше всего сделать это в разделе "Реализация". Также вы сможете использовать "Влияние кода" для того, чтобы помочь вам понять код.

1. Внедрение в проектный путь дизайна, ориентированного на объект, не так просто, и для повторного использования его становится еще ____.

- a. детализации
- b. труднее
- c. повторного
- d. на

2. Необходимо выбирать подходящие объекты, относить их к различным классам, сохранять их ____ степень детализации.

- a. момента
- b. трудным
- c. разумную
- d. другому

3. Проектирование должно, с одной стороны, быть совместимым с задачей, которую необходимо выполнить, и быть общим для удовлетворения всех требований, ____ могут возникнуть в будущем.

- a. государство
- b. декоратор
- c. также

d. которые

4. Однако проектировщики могут создать хорошую систему проектирования, хотя они шокированы _____, что могут предложить ряд вариантов.?

a. выполнение

b. делать

c. она

d. тем

5. Сценариям нелегко придумать совершенно новую историю, вместо этого они берут за основу шаблоны, жанры и изображения, которые _____ разработаны в мировой литературе.?

a. поведение

b. для

c. нуждается

d. были

6. Мотивом убийства являются деньги, _____, ревность и т.д.?

a. при

b. решение

c. месть

d. наследование

7. Из этой книги вы не _____, как создавать пользовательские интерфейсы, как писать устройства и как работать с базами данных, ориентированными на объект.?

- a. знаете
- b. которые
- c. подклассом
- d. это

8. Кроме того, отсутствует информация о жалких людях, характерных для конкретных _____ областей.?

- a. тематических
- b. это
- c. большинство
- d. один

9. В каждой из этих областей могут _____ их собственные жалкие устройства, и, возможно, кто-то их систематизировал.?

- a. существовать
- b. и
- c. тип
- d. уменьшают

10. В дизайне, ориентированном на _____, повторное использование часто является важным фактором.?

- a. дизайнером
- b. объект
- c. ненужной
- d. содержится

11. Результаты ____ результатом применения жалкого и различного рода компромиссов.?

- a. такая
- b. тот
- c. являются
- d. существующей

12. Перерасчет всех эффектов ____ вам понять и оценить их роль.?

- a. вы
- b. является
- c. поможет
- d. каждой

13. В наших патологиях он будет ____ возможности Smalltalk и C++.

- a. замену
- b. преимущества
- c. использовать
- d. случае

14. В языке CLOS есть несколько методов, которые делают невозможной ____ посетителя ненужной.?

- a. модель
- b. не
- c. композитного
- d. общим

15. MVC поддерживает ____ вид с помощью класса ComputerView, который является подклассом класса Вид.?

- a. внутреннего
- b. упростить
- c. и
- d. встроенный

16. ____ одна характеристика MVC заключается в том, что этот вид может быть включен.?

- a. лучше
- b. еще
- c. алгоритма
- d. кто

17. ____, допускается представление контрольной панели, состоящей из кнопок, в качестве композитного обзора, содержащего шаги, с помощью одной кнопки на каждой из них.?

- a. например
- b. за
- c. метеодов
- d. такие

18. ICC использует другие ____ дизайна, такие как производственный метод для установления класса контрольного устройства по умолчанию для просмотра, и декоратор для добавления к виду вращаемости.?

- a. предполагаемого

- b. приводит
- c. паттеры
- d. вращаемости

19. Базовая связь в _____ ICC описывается parter в наблюдателях, компостере и стратегии.?

- a. объекты
- b. отношении
- c. может
- d. схеме

20. Стратегия является _____ для представления алгоритма.?

- a. объектом
- b. затронуты
- c. две
- d. к

21. Она полезна, когда вы хотите статическую или динамическую замену одного алгоритма на другой или когда алгоритм связан со сложными структурами _____.?

- a. данных
- b. к
- c. а
- d. в

22. В приложениях содержится информация, _____ поможет вам лучше понять путь и связанные с этим вопросы.?

- a. с
- b. которая
- c. постоянными
- d. эти

23. Пример _____ фрагментации, который иллюстрирует вероятность осуществления на языках C++ или Smalltalk.?

- a. кода
- b. ориентированного
- c. многих
- d. уменьшают

24. Результаты _____, компромиссы, которые должны быть сделаны.?

- a. мало
- b. эта
- c. тип
- d. приложения

25. Какие аспекты поведения системы _____ быть независимо изменены с помощью этого типа??

- a. зависит
- b. могут
- c. зачастую
- d. базами

26. Обычная (команда) _____ запрос как объект.?

- a. инкапсулирует
- b. запросы
- c. изменены
- d. замену

27. Группирует объекты в структуры, похожие на деревья, чтобы представлять иерархический тип "serv-единица" позволяет ____ работать с отдельными объектами в том же состоянии, что и набор объектов.?

- a. и
- b. клиентам
- c. может
- d. декоратор

28. Декораторы используются для расширения ____ функциональности.?

- a. название
- b. как
- c. существующей
- d. этапе

29. Государство (состояние) (291) допускает изменение поведения объекта при изменении его ____ состояния. "?

- a. этот
- b. в
- c. внутреннего
- d. состоящей

30. _____ " (Стратегия) (300) определяет положение семейства алгоритмов, инкапсулирует все из них и позволяет заменить одно из них другим. "?"

- a. причин
- b. dmt
- c. именно
- d. страттаги

31. Посетитель " (314) представляет операцию, которая должна быть проведена над _____ объекта.?

- a. элементами
- b. эффектов
- c. позволяет
- d. при

32. Только те, кто концентрируется только на межклассовых отношениях, считаются в категории "_____ патометров", отмечают, что большинство пасторов работают на уровне объектов.?

- a. классовых
- b. фактором
- c. для
- d. класс

33. Поведение классов _____ наследство для описания алгоритмов и потока управления.?

- a. определяемым
- b. совершенно

c. вместе

d. использует

34. Поведение объектов описывает _____ как _____, принадлежащие к определенной группе.?

a. описывается

b. запроса

c. между

d. объекты

35. Наиболее трудная задача в _____, ориентированном на объект, заключается в том, чтобы поместить систему в объекты.?

a. на

b. такие

c. проектировании

d. в

36. Интерфейс описывает все запросы, которые _____ быть отправлены на объект.?

a. по

b. в

c. могут

d. дизайне

37. Тип - это название, используемое для _____ конкретного интерфейса.?

a. интерфейса

- b. и
- c. в
- d. указания

38. Другие модели указывают способ разложения объекта на более мелкие _____.?

- a. оба
- b. различным
- c. на
- d. подобъекты

39. Абстрактный субобъект и строитель описывают объекты, единственная цель которых - _____ другие объекты.?

- a. создать
- b. понимается
- c. потока
- d. smalltalk

40. _____ и команда - это объекты, ответственные за выполнение запроса на другой объект или группу.?

- a. посетитель
- b. тип
- c. трудно
- d. возможно

41. _____ модели показывает, как представить целую подсистему в качестве объекта.?

- a. приводит
- b. факад
- c. часто
- d. используемое

42. _____ модели показывает, как поддерживать большое число объектов с высокой степенью детализации.?

- a. гибким
- b. адаптер
- c. абстрактных
- d. теперь

43. Динамическая связывание _____, что отправка определенной просьбы не определяет какой-либо конкретной реализации до момента выполнения.?

- a. сделать
- b. означает
- c. к
- d. такие

44. Различные объекты, поддерживающие один и тот же интерфейс, могут реагировать на _____ запросы различными операциями.?

- a. такие
- b. повторного
- c. того
- d. программы

45. Такая эксплуатационная совместимость объектов называется полиморфизмом и является важной особенностью объектно-ориентированных _____.?

- a. объектом
- b. определен
- c. не
- d. систем

46. Она позволяет клиенту не делать практически никаких предположений в отношении объектов, за исключением того, что они поддерживают определенный объект, что сам полиморфизм может быть определен объектом и что он может быть принят решение о его _____ в том же состоянии.?

- a. выполнении
- b. является
- c. состоит
- d. быть

47. _____ указывает внутренние данные объекта и его представление.?

- a. аптерны
- b. похожие
- c. при
- d. класс

48. приложение В), класс может быть представлен в виде _____, в котором название класса написано жирным шрифтом.?

- a. и
- b. прямоугольника

c. быть

d. тематической

49. Любые ____, определенные для класса, следуют после операций.?

a. применения

b. данные

c. истинное

d. рекомендуемые

50. Эта ____ может представлять собой псевдокод, описывающий выполнение операции.?

a. диаграмма

b. описывается

c. состоящей

d. системы

51. В таком ____ код представляется в прямоугольнике со сложным углом, соединенным пунктирной линией с операцией, которую он выполняет.?

a. случае

b. является

c. компиляции

d. площадь

52. Класс объекта называется классом (класс миксина), цель которого состоит в обеспечении дополнительного ____ или функциональности для других классов.?

a. интерфейса

- b. пейзажем
- c. получает
- d. момент

53. Название абстрактных классов курсивом отличается от ____.

- a. конкретных
- b. сделать
- c. объект
- d. знаниями

54. Заголовок также ____ для указания абстрактных операций.

- a. но
- b. иллюстрирует
- c. используется
- d. существующего

55. В Smalltalk наследование понимается только ____ последовательность реализации.

- a. возможно
- b. как
- c. а
- d. строго

56. Программное обеспечение C++ манипулирует объектами по типам, определяемым абстрактными ____.

- a. классом

- b. для
- c. алгоритм
- d. классами

57. ____ наследование интерфейса может быть применено к C++ с помощью открытого наследства абстрактному классу.?

- a. истинное
- b. реализации
- c. установлен
- d. не

58. В Smalltalk ____, построенных в соответствии с жалкой цепью обязанностей, должен иметь общий тип, но их реализация, как правило, является иной.?

- a. быть
- b. объектов
- c. найти
- d. алгоритмов

59. Существует два преимущества ____ объектами строго через абстрактный интерфейс класса.?

- a. на
- b. разделе
- c. на
- d. манипулирования

60. Клиент не нуждается в информации о конкретных типах ____, которые он

использует, при условии, что все они имеют ожидаемый интерфейс.?

- a. внимание
- b. объектов
- c. а
- d. между

61. Эти преимущества значительно сокращают количество связей между _____.?

- a. быть
- b. подсистемами
- c. над
- d. классами

62. Возможно даже сформулировать принцип объектно-ориентированного _____ для повторного использования.?

- a. другому
- b. число
- c. типами
- d. дизайна

63. Не _____ переменные в этом месте в качестве копий конкретных классов.?

- a. этапе
- b. подход
- c. объект
- d. объявляйте

64. Вместо этого держитесь интерфейса, определенного _____.?

- a. классом
- b. и
- c. просмотра
- d. в

65. Его легче использовать, поскольку он _____ поддерживается языком программирования.?

- a. непосредственно
- b. в
- c. однако
- d. помочь

66. В случае чередования классов также упрощается задача _____ существующего осуществления.?

- a. картеров
- b. определенного
- c. изменения
- d. объектом

67. Если подкласс заменяет только _____ операции, то могут быть затронуты другие унаследованные операции, поскольку они могут вызвать замену.?

- a. со
- b. имеют
- c. класс
- d. некоторые

68. Однако подкласс и материнский класс настолько тесно связаны между собой, что любое изменение в последнем _____ изменения в осуществлении подкласса.?
- a. потребует
 - b. computerview
 - c. степени
 - d. практически
69. По сравнению с составом объектов, а не последовательностью классов, вы _____ в себя каждый класс и позволяете ему выполнять только свою задачу.?
- a. для
 - b. для
 - c. класс
 - d. включаете
70. Классы и их иерархия _____ небольшими, и вероятность их роста до неуправляемого размера невелика.?
- a. небольшими
 - b. отличительной
 - c. ориентированном
 - d. остаются
71. Конструкция, основанная на составе, будет содержать больше объектов (хотя число классов может уменьшиться), и поведение _____ будет зависеть от их взаимодействия.?
- a. системы
 - b. определенный

c. методом

d. дизайна

72. Теперь класс Windows может четко перенаправить запросы на свой _____ Rectangle, а не наследовать его операции.?

a. расширения

b. член

c. задачи

d. и

73. Внутренняя площадь окна может быть круглой во время реализации, просто путем предоставления вместо этого копии _____ Rectangle.?

a. со

b. класса

c. одного

d. в

74. Предполагается, что оба класса имеют _____ и тот же тип конструкции.?

a. один

b. в

c. информация

d. называется

75. Делегация также является отличительной особенностью места и _____ подходов, применяемых для повышения гибкости в отношении состава объектов.?

- a. объекта
- b. инкапсулирует
- c. повторного
- d. других

76. Конструкция не одинакова, один и тот же тип _____.?

- a. дизайна
- b. и
- c. от
- d. более

77. Другие _____, которые должны применяться для разработки и другого подхода, чтобы сделать его более гибким и сделать его более гибким.?

- a. связан
- b. понять
- c. подходы
- d. тип

78. При наличии существенного сходства между абстракцией и конкретным осуществлением абстракция может просто _____ операции.?

- a. способность
- b. делегировать
- c. функциональности
- d. класса

79. Наследование и параметризованные типы являются другим (хотя и не

совсем объективным) _____ повторного использования имеющихся функциональных возможностей.?

- a. жанры
- b. быть
- c. стороны
- d. методом

80. Этот метод позволяет идентифицировать тип _____ указания типов, которые он использует.?

- a. конкретных
- b. без
- c. и
- d. не

81. Неконкретные типы элементов представлены в _____ характеристик, имеющих в опционах.?

- a. полезна
- b. перечне
- c. подклассами
- d. на

82. Структуры времени и времени программы, ориентированной на объект, на _____ осуществления зачастую имеют мало общего со структурой ее исходного кода.?

- a. конструкционного
- b. осуществлении

c. различные

d. стадии

83. Последний установлен на этапе компиляции; код состоит из классов, взаимоотношения _____ не являются постоянными.?

a. по

b. и

c. на

d. которых

84. Можно обеспечить выполнение по умолчанию, которое может _____ заменено подклассами.?

a. от

b. быть

c. в

d. исключением

85. Однако ни наследование, ни параметризованные типы не могут быть _____ на языке, таком, как Smalltalk.?

a. используются

b. исполнение

c. осуществления

d. изменены

86. Различие между знаниями и агрегированием, скорее, зависит от предполагаемого использования, а не от языковых _____.?

a. механизмов

b. и

c. объекты

d. и

87. Таким образом, в структуре этапа компиляции не так _____ увидеть разницу, но она тем не менее значительна.?

a. содержится

b. просто

c. которые

d. то

88. Взаимосвязь между объектами и их типами должна быть очень тщательной, поскольку именно введение в их конструкцию зависит от степени успеха _____ и провала структуры на этапе осуществления.?

a. жирным

b. документально

c. дизайна

d. о

89. Поведение системы на момент внедрения должно определяться дизайнером, а не _____.?

a. абстракцией

b. со

c. копии

d. языком

90. Реконструирование системы может _____ трудным, и дизайн объекта может _____ трудным.?

- a. определяет
- b. описывает
- c. быть
- d. может

91. Вот некоторые из типичных _____ для перепроектирования, а также патологии, которые избегают этого.?

- a. причин
- b. показывает
- c. объекта
- d. для

92. При создании объекта явно _____ класс.?

- a. исходного
- b. которую
- c. iss
- d. существует

93. При _____ имени класса вы связываете себя с конкретной реализацией, а не с интерфейсом.?

- a. с
- b. хотя
- c. присвоении

d. член

94. Это может затруднить изменение _____ в будущем.?

a. вы

b. объекта

c. из

d. вовсе

95. Чтобы избежать такой проблемы, _____ объекты косвенно.?

a. получает

b. создайте

c. могут

d. интерфейс

96. Устанавливая конкретную операцию, вы _____ себя единственным способом выполнения запроса.?

a. включает

b. ограничиваете

c. о

d. которые

97. Если вы не _____ запросы в код, то будет легче изменить способ реагирования на запрос как на этапе компиляции, так и на этапе выполнения.?

a. быть

b. из

c. детализации

d. включаете

98. Создание подклассов приводит к _____ увеличению числа классов.?

a. общему

b. мотивом

c. путь

d. скорее

99. Для расширения внутреннего повторного использования будут использоваться картеры, которые уменьшают _____ взаимоотношений.?

a. для

b. количество

c. перепроектирования

d. создавать

100. Более низкая степень подключения также повышает способность системы _____.?

a. могут

b. развивать

c. классами

d. включен

101. Расширение класса легче, если _____ не зависит от многих других.?

a. оно

b. ориентированные

c. устройства

d. to

102. Часто оно включает классы из одной или нескольких библиотек заранее _____ классов.?

a. представить

b. должны

c. определенных

d. взаимоотношений

103. В рамках этой программы разрабатываются общие для данной тематической области _____ по проектированию, основное внимание в рамках которой уделяется повторному использованию дизайна, хотя и включает в себя конкретные подклассы, которые могут применяться непосредственно.?

a. представлять

b. решения

c. качестве

d. них

104. _____ использование на этом уровне изменяет взаимосвязь между приложением и его программным обеспечением.?

a. повторное

b. относить

c. инкапсулирует

d. не

105. Третье _____ упростить разработку приложений для финансового

моделирования.?

- a. дизайна
- b. в
- c. объектов
- d. позволит

106. Рамки, в которых они решаются с помощью применения патана, могут лучше обеспечить высокий уровень _____ и повторного использования кода.?

- a. между
- b. подходы
- c. любое
- d. дизайна

107. В использованных _____ можно найти несколько различных картеров дизайна.?

- a. изменить
- b. шаблоны
- c. каркатурах
- d. программное

108. Структуры помогают адаптировать архитектуру _____ каркаупатов к различным видам применения.?

- a. создать
- b. солености
- c. чтобы
- d. мотивом

109. Дополнительное преимущество заключается в том, _____ вместе с рамками эти патоны, которые в них используются, документально зафиксированы [B]94].?

- a. чтобы
- b. быть
- c. стратегия
- d. что

110. Системы, ориентированные на _____, могут быть в максимально возможной степени повторно использованы.?

- a. использования
- b. имеют
- c. языком
- d. объект

111. Крупные прикладные программы, ориентированные на _____, получают из слоев, которые взаимодействуют друг с другом.?

- a. объект
- b. компиляции
- c. делать
- d. команда

112. Конструкция и код приложения в _____ степени получены от машин, применявшихся при его создании.?

- a. различными
- b. программным

c. значительной

d. класса

113. Если дизайнер получает каталог из более ____ 20 пасторов, то трудно решить, какой тип наиболее подходит для конкретной проектировочной задачи.?

a. единственный

b. литературе

c. чем

d. тип

114. В таблице 1.2 перечислены те аспекты ____, которые различаются от одного жалкого к другому.?

a. конструкции

b. к

c. в

d. конкретной

115. Этот подход не ____ изучению причин, приведших к необходимости реорганизации.?

a. быть

b. которые

c. способствует

d. для

116. Вместо этого, ____ о том, что может изменить дизайн и что вы хотели бы изменить без перепроектирования.?

- a. какой
- b. подумайте
- c. становится
- d. что

117. Со временем вы будете _____ свой собственный подход к работе с пейзажем.?

- a. фрагментации
- b. ему
- c. в
- d. разрабатывать

118. Невозможно будет использовать общий метод дизайна заводского дизайнера _____ сайта.?

- a. прототипа
- b. приведших
- c. в
- d. важной

119. Способность _____, вероятно, изменит способность создавать путь.?

- a. аптерны
- b. постоянными
- c. один
- d. делегировать

120. Средства _____ единого объекта подкласса конструкционного объекта.?

- a. создания

- b. многих
- c. поведения
- d. наследование

121. Возможно изменить поведение пути поведения как путь к посадке, путь пути или путь, _____ вовсе не является дорожкой.?

- a. моделирования
- b. язык
- c. который
- d. многих

122. Вы _____ совет о том, как лучше всего сделать это в разделе "Реализация".?

- a. имени
- b. найдете
- c. же
- d. перечне

123. Также вы сможете использовать "Влияние кода" для _____, чтобы помочь вам понять код.?

- a. в
- b. которых
- c. них
- d. того