Вопрос №1

|  |  |
| --- | --- |
|  | **Что представляет собой CASE-технологии?** |
| 1 | Методологияпроектирования ИС |
| 0 | Новые технологии |
| 0 | Интернет технологии |
| 0 | Стандартная библиотека |
| 0 | Методологияпроектирования ОС |

Вопрос №2

|  |  |
| --- | --- |
|  | **В виде чего чаще всего хранится образ программы?** |
| 1 | Исполняемого модуля |
| 0 | Неисполняемого модуля |
| 0 | Основного компонента ПО |
| 0 | Тест - кейса |
| 0 | Временного файла |

Вопрос №3

|  |  |
| --- | --- |
|  | **На сколько групп делятся компьютерные пользователи по отношению к ПО?** |
| 1 | 3 |
| 0 | 4 |
| 0 | 1 |
| 0 | 2 |
| 0 | 5 |

Вопрос №4

|  |  |
| --- | --- |
|  | **Программа, упорядоченная в соответствии с некоторым алгоритмом последовательности команд компьютера для решения задачи пользователя - это..?** |
| 1 | Основной компонент ПО |
| 0 | Основной компонент ОС |
| 0 | Исполняемый модуль |
| 0 | Кейс |
| 0 | Временный файл |

Вопрос №5

|  |  |
| --- | --- |
|  | **Лицо, заинтересованное в решении некоторой задачи средствами вычислительной техники - это** |
| 1 | Пользователь |
| 0 | Системный программист |
| 0 | Прикладной программист |
| 0 | Руководитель проекта |
| 0 | Технический менеджер |

Вопрос №6

|  |  |
| --- | --- |
|  | **Какой термин в программировании означает единицу работы вычислительной системы, требующую выделения вычислительных ресурсов?** |
| 1 | Задача |
| 0 | Проект |
| 0 | Тест кейс |
| 0 | Приложение |
| 0 | Программа |

Вопрос №7

|  |  |
| --- | --- |
|  | **Программная реализация решения задачи на компьютере - это** |
| 1 | Приложение |
| 0 | Задача |
| 0 | Алгоритм |
| 0 | Кейс |
| 0 | Модуль |

Вопрос №8

|  |  |
| --- | --- |
|  | **Что может служить примером большой программы?** |
| 1 | Стандартная библиотека Java или C# |
| 0 | Функции |
| 0 | Тест кейс |
| 0 | CMS |
| 0 | WEB - сайт |

Вопрос №9

|  |  |
| --- | --- |
|  | **Сколько этапов содержит в себе процесс создания программ?** |
| 1 | 3 |
| 0 | 2 |
| 0 | 4 |
| 0 | 5 |
| 0 | 6 |

Вопрос №10

|  |  |
| --- | --- |
|  | **Точная формулировка требований предъявляемых к работе программы - это?** |
| 1 | Постановка задачи |
| 0 | Техническое задание |
| 0 | Документация |
| 0 | Стандартная библиотека |
| 0 | Постановка условия |

Вопрос №11

|  |  |
| --- | --- |
|  | **Алгоритм представляет собой** |
| 1 | Точный набор инструкций |
| 0 | Техническое задание |
| 0 | Документация |
| 0 | Описание методов |
| 0 | Процесс создания программы |

Вопрос №12

|  |  |
| --- | --- |
|  | **Какой характер носит разработка ПС?** |
| 1 | Творческий |
| 0 | Свободный |
| 0 | Серьезный |
| 0 | Официальный |
| 0 | Деловой |

Вопрос №13

|  |  |
| --- | --- |
|  | **В чем главное различие между технологией программирования и программной инженерией как дисциплинами для изучения?** |
| 1 | Способ рассмотрения и систематизации материала |
| 0 | Процесс создания программы |
| 0 | Документация |
| 0 | Описание методов |
| 0 | Описание постановки задачи |

Вопрос №14

|  |  |
| --- | --- |
|  | **На что делается акцент в технологии программирования?** |
| 1 | Изучение процессов разработки ПС |
| 0 | Способ рассмотрения и систематизации материала |
| 0 | Описание ТЗ |
| 0 | Описание постановки задачи |
| 0 | Документация |

Вопрос №15

|  |  |
| --- | --- |
|  | **Как в методологии программирования рассматриваются методы?** |
| 1 | Снизу |
| 0 | Сверху |
| 0 | По этапам |
| 0 | По техническому заданию |
| 0 | Нет верного варианта |

Вопрос №16

|  |  |
| --- | --- |
|  | **Как в технологии программирования рассматриваются методы?** |
| 1 | Сверху |
| 0 | Снизу |
| 0 | По этапам |
| 0 | По техническому заданию |
| 0 | Нет верного варианта |

Вопрос №17

|  |  |
| --- | --- |
|  | **Сколько этапов можно выделить в историческом аспекте в развитии технологии программирования?** |
| 1 | 3 |
| 0 | 2 |
| 0 | 1 |
| 0 | 4 |
| 0 | 5 |

Вопрос №18

|  |  |
| --- | --- |
|  | **Какой компанией были разработаны основы компонентного подхода?** |
| 1 | Microsoft |
| 0 | OLE |
| 0 | DCOM |
| 0 | Github |
| 0 | ActiveX |

Вопрос №19

|  |  |
| --- | --- |
|  | **На основе чего была создана технология ActiveX?** |
| 1 | OLE - automation |
| 0 | MDS |
| 0 | MTS |
| 0 | Microsoft |
| 0 | CORBA |

Вопрос №20

|  |  |
| --- | --- |
|  | **В основе какой технологии лежит компонентный подход?** |
| 1 | CORBA |
| 0 | MDS |
| 0 | MTS |
| 0 | Microsoft |
| 0 | OLE - automation |

Вопрос №21

|  |  |
| --- | --- |
|  | **Традиционный метод борьбы со сложностью?** |
| 1 | Метод «Разделяй и властвуй» |
| 0 | "влезал в голову" |
| 0 | Не накручивайте себя |
| 0 | Правильный настрой |
| 0 | Все ответы верные |

Вопрос №22

|  |  |
| --- | --- |
|  | **1 уровень архитектуры ПО?** |
| 1 | Центральный Диспетчер |
| 0 | Местные диспетчеры |
| 0 | Функциональные программы |
| 0 | Стандартные программы |
| 0 | Все ответы верные |

Вопрос №23

|  |  |
| --- | --- |
|  | **Архитектуре программного обеспечения как иерархической системе присущ свойства?** |
| 0 | вертикальная соподчиненность, заключающаяся в последовательном упорядоченном расположении взаимодействующих компонент, составляющих данный комплекс программ; |
| 0 | компоненты одного уровня обеспечивают реализацию функций компонент следующего уровня; |
| 0 | каждый уровень иерархии реализуется через функции ком­понент более нижних уровней; |
| 0 | каждый компонент знает о компонентах более низких уровней и ничего не знает о компонентах более высоких уровней; |
| 1 | Все ответы верные |

Вопрос №24

|  |  |
| --- | --- |
|  | **2 уровень Архитектуры ИС?** |
| 1 | ИТ-архитектуры |
| 0 | Бизнес-архитектура |
| 0 | Программная архитектура |
| 0 | Архитектура данных |
| 0 | Все ответы верные |

Вопрос №25

|  |  |
| --- | --- |
|  | **Аспект ИТ-архитектуры?** |
| 1 | Среда, обеспечивающая реализацию бизнес-приложений |
| 0 | Определяет бизнес-стратегии, управление, организацию |
| 0 | Описание отдельного приложения |
| 0 | Компьютерная программа |
| 0 | Все ответы верны |

Вопрос №26

|  |  |
| --- | --- |
|  | **Подразделения ИС на классы?** |
| 1 | Все ответы верные |
| 0 | Системы на основе архитектуры файл-сервер |
| 0 | Системы на основе архитектуры клиент-сервер |
| 0 | Системы на основе многоуровневой архитектуры |
| 0 | Системы на основе интернет/интранет-технологий |

Вопрос №27

|  |  |
| --- | --- |
|  | **Первый функциональный компонент ИС?** |
| 1 | Пользовательский интерфейс |
| 0 | Бизнес-логика |
| 0 | Управление данными |
| 0 | Логика управления |
| 0 | Все ответы верные |

Вопрос №28

|  |  |
| --- | --- |
|  | **Укажите что относится к спиральной модели** |
| 1 | Все ответы верные |
| 0 | На первых этапах уточняются спецификации продукта |
| 0 | На последующих этапах добавляются новые возможности и функции |
| 0 | По окончании каждой этерации осуществляют оценку рисков продолжения работ |
| 0 | Разработка приложения выглядит как серия последовательных итераций |

Вопрос №29

|  |  |
| --- | --- |
|  | **Недостатки архитектуры файл-сервер?** |
| 1 | Все ответы верны |
| 0 | трудности при обеспечении непротиворечивости и целостности данных; |
| 0 | существенная загрузка локальной сети передаваемыми данными; |
| 0 | в целом, невысокая скорость обработки и представления информации; |
| 0 | высокие требования к ресурсам компьютеров. |

Вопрос №30

|  |  |
| --- | --- |
|  | **Объект разработки ИС** |
| 1 | диалог и логики обработки |
| 0 | компьютер-сервер, содержащий корпоративную и персональную базы; |
| 0 | компьютер-сервер и персональные компьютеры с ПБД; |
| 0 | несколько компьютеров-серверов и персональных компьютеров с ПБД. |
| 0 | Все ответы верные |

Вопрос №31

|  |  |
| --- | --- |
|  | **Достоинства архитектуры клиент-сервер** |
| 1 | Независимость данных от прикладных программ, приложение, работающее на рабочей станции, не читает записи базы данных "напрямую", а посылает запросы на сервер, где они принимаются и последовательно отрабатываются специальными программами. |
| 0 | Сложные приложения, вызывающие большое взаимодействие с БД, сильно загружают как клиента, так и сеть. |
| 0 | Результаты SQL-запроса должны вернуться клиенту для обработки, потому что там находится логика принятия решения. |
| 0 | Такая схема приводит к дополнительному усложнению администрирования приложений, разбросанных по различным клиентским узлам |
| 0 | Все ответы верные |

Вопрос №32

|  |  |
| --- | --- |
|  | **Верное определение хранимых процедур** |
| 1 | Все ответы верные |
| 0 | это процедура с операторами SQL для доступа к БД, вызываемая по имени с передачей требуемых параметров и выполняемая на сервере БД. |
| 0 | Хранимые процедуры могут компилироваться, что повышает скорость их выполнения и сокращает нагрузку на сервер. |
| 0 | Хранимые процедуры улучшают целостность приложений и БД, гарантируют актуальность коллективно используемых операций и вычислений. |
| 0 | Улучшается сопровождение таких процедур, а также безопасность (нет прямого доступа к данным). |

Вопрос №33

|  |  |
| --- | --- |
|  | **Клиент-сервер (двухуровневые системы) недостаток** |
| 1 | Излишнее утяжеление хранимых процедур прикладной логикой может перегрузить сервер, что приведет к потере производительности. |
| 0 | Повышают эффективность выполнения приложений за счет использования возможностей сервера БД, разгрузки сети и обеспечения контроля целостности данных. |
| 0 | Архитектура клиент-сервер получила широкое распространение как способ организации приложений для рабочих групп и информационных систем корпоративного уровня. |
| 0 | схема позволяет получить более сбалансированную загрузку клиентов и сервера, но при этом затрудняется сопровождение приложений. |
| 0 | Все ответы верные |

Вопрос №34

|  |  |
| --- | --- |
|  | **Нижний уровень многоуровневой архитектуры это -** |
| 1 | приложения клиентов, выделенные для выполнения функций и логики представлений PS и PL и имеющие программный интерфейс для вызова приложения на среднем уровне; |
| 0 | это сервер приложений, на котором выполняется прикладная логика BL и с которого логика обработки данных DL вызывает операции с базой данных DS; |
| 0 | находится удаленный специализированный сервер базы данных, выделенный для услуг обработки данных DS и файловых операций FS (без риска использования хранимых процедур). |
| 0 | Без уровней |
| 0 | Все ответы верные |

Вопрос №35

|  |  |
| --- | --- |
|  | **Достоинство многоуровневой архитектуры** |
| 1 | Все ответы верны |
| 0 | Трехуровневая архитектура позволяет сбалансировать нагрузку на разные узлы и сеть, а также способствует специализации инструментов для разработки приложений. |
| 0 | Централизация логики приложения упрощает администрирование и сопровождение. |
| 0 | Четко разделяются платформы и инструменты для реализации интерфейса и прикладной логики. |
| 0 | Изменения прикладной логики не затрагивают интерфейса, и наоборот. |

Вопрос №36

|  |  |
| --- | --- |
|  | **Верные определения архитектуры веб-приложения** |
| 1 | Все ответы верные |
| 0 | В развитии технологии Интернет/интранет основной акцент пока что делается на разработке инструментальных программных средств. |
| 0 | наблюдается отсутствие развитых средств разработки приложений, работающих с БД. |
| 0 | Компромиссным решением для создания удобных и простых в использовании и сопровождении БД, эффективно работающих с БД, стало объединение Интернет/интранет-технологии с многоуровневой архитектурой. |
| 0 | Интеграция Интернет/интранет-технологий и архитектуры клиент- сервер процесс внедрения корпоративной ИС но упрощается при сохранении достаточно высокой эффективности и простоты совместного использования информации |

Вопрос №37

|  |  |
| --- | --- |
|  | **Верное определение** |
| 1 | Процесс проектирования архитектуры программного обес­печения состоит в проектировании структуры всех его ком­понент, функционально связанных с решаемой задачей, включая сопряжения между ними и требования к ним |
| 0 | Архитектура программного обеспечения в традиционном смысле включает определение только часть модулей программ, их иерархии и сопряжения между ними и данными. |
| 0 | Если разрабатывается отдельная программа, исходными дан­ными для этого процесса будут начальные концепции. |
| 0 | Если разрабатывается система программного изделия, исход­ными данными для этого процесса будут архитектуры системы. |
| 0 | Все ответы верны |

Вопрос №38

|  |  |
| --- | --- |
|  | **Стремление к созданию модульных программ объясняется следующими факторами:** |
| 1 | Все ответы верны |
| 0 | программные модули, как правило, решают небольшую функциональную задачу, используют на входе и на выходе немного данных. |
| 0 | модульные программы легко читать, сопровождать и мо­дифицировать. |
| 0 | модульные программы обладают повышенной надеж­ностью, так как при их разработке существует возможность распределения работ по созданию модулей различной сложности и важности между программистами различной квалификации; |
| 0 | процедура загрузки всей программы в оперативную память упрощается при использовании метода оверлейности |

Вопрос №39

|  |  |
| --- | --- |
|  | **Укажите критерий хорошей архитектуры имеющий следующее определение:**  **Возможность добавлять в систему новые сущности и функции, не нарушая ее основной структуры.** |
| 1 | Расширяемость системы |
| 0 | Эффективность системы |
| 0 | Гибкость системы |
| 0 | Масштабируемость системы |
| 0 | Тестируемост системы |

Вопрос №40

|  |  |
| --- | --- |
|  | **Укажите критерий хорошей архитектуры имеющий следующее определение:**  **Возможность сократить срок разработки за счёт добавления к проекту новых людей** |
| 1 | Масштабируемость системы |
| 0 | Эффективность системы |
| 0 | Гибкость системы |
| 0 | Расширяемость системы |
| 0 | Тестируемост системы |

Вопрос №41

|  |  |
| --- | --- |
|  | **Программное обеспечение это?** |
| 1 | Совокупность программ, позволяющих организовать решение задачи на ЭВМ. |
| 0 | Универсальное устройство для передачи информации |
| 0 | Операционные системы |
| 0 | Устройство для хранения информации |
| 0 | Оперативно запоминающее устройство |

Вопрос №42

|  |  |
| --- | --- |
|  | **Системное ПО предназначено для?** |
| 1 | Обслуживания самого компьютера, для управления работой его устройств. |
| 0 | Выполнения вычислительных процессов |
| 0 | Хранения и вывода информации |
| 0 | Проверки работоспособности устройств |
| 0 | Для запоминания информации на внешних устройствах |

Вопрос №43

|  |  |
| --- | --- |
|  | **К программному обеспечению относится:** |
| 1 | Все перечисленное. |
| 0 | Технология проектирования программ |
| 0 | Методы тестирования программ |
| 0 | Методы доказательства правильности программ |
| 0 | Анализ качества работы программ |

Вопрос №44

|  |  |
| --- | --- |
|  | **Чем определяется сфера применения конкретного компьютера?** |
| 1 | Программным обеспечением |
| 0 | Интерфейсом |
| 0 | Оперативной памятью |
| 0 | Процессором |
| 0 | Количеством ядер |

Вопрос №45

|  |  |
| --- | --- |
|  | **Что относится к программному обеспечению современных компьютеров?** |
| 1 | Все перечисленное |
| 0 | Игры |
| 0 | Браузеры |
| 0 | Редакторы |
| 0 | Антивирусы |

Вопрос №46

|  |  |
| --- | --- |
|  | **На какие группы можно разделить ПО современных компьютеров?** |
| 1 | Все кроме вредоносного ПО |
| 0 | Системное ПО |
| 0 | Прикладное ПО |
| 0 | Аппаратное обеспечение |
| 0 | Вредоносное ПО |

Вопрос №47

|  |  |
| --- | --- |
|  | **Виды технологических этапов процесса проектирования?** |
| 1 | Кодирование, отладка, тестирование |
| 0 | Кодирование, тестирование, запуск |
| 0 | Кодирование, отладка, продажа |
| 0 | Кодирование, добавление комментариев, тестирование |
| 0 | Кодирование, отладка, запуск |

Вопрос №48

|  |  |
| --- | --- |
|  | **Кто осуществляет непосредственную эксплуатацию программных продуктов?** |
| 1 | Потребители |
| 0 | Программисты |
| 0 | Дизайнеры |
| 0 | Маркетологи |
| 0 | Проект менеджеры |

Вопрос №49

|  |  |
| --- | --- |
|  | **Что такое ППП?** |
| 1 | Пакет прикладных программ |
| 0 | Пятиразовое правильное питание |
| 0 | Проектирование приклдных программ |
| 0 | Пакет прикладных проектов |
| 0 | Прикладные программные продукты |

Вопрос №50

|  |  |
| --- | --- |
|  | **Кем выполняется установка ППП?** |
| 1 | Системным администратором |
| 0 | Дизайнером |
| 0 | Менеджером кол центра |
| 0 | Маркетологом |
| 0 | Проект менеджером |

Вопрос №51

|  |  |
| --- | --- |
|  | **Что входит в состав системного ПО?** |
| 1 | Все варианты верны |
| 0 | Операционная система |
| 0 | Трансляторы языков |
| 0 | Обслуживающие программы |
| 0 | Все варианты не верны |

Вопрос №52

|  |  |
| --- | --- |
|  | **Виды программного обеспечения ЭВМ?** |
| 1 | Все варианты верны |
| 0 | Системное ПО |
| 0 | Инструментальное ПО |
| 0 | Прикладное ПО |
| 0 | Все варианты не верны |

Вопрос №53

|  |  |
| --- | --- |
|  | **Классы инструментального ПО?** |
| 1 | Все варианты верны |
| 0 | Языки и системы программирования |
| 0 | Интегрированные среды программирования |
| 0 | Программные комплексы |
| 0 | Все варианты не верны |

Вопрос №54

|  |  |
| --- | --- |
|  | **Виды сервисного ПО?** |
| 1 | Все перечисленное |
| 0 | Поисковые в сети |
| 0 | Диагностика |
| 0 | Антивирусные программы |
| 0 | Архиваторы |

Вопрос №55

|  |  |
| --- | --- |
|  | **Классы базового ПО?** |
| 1 | Все варианты верны |
| 0 | Операционные системы |
| 0 | Оболочки ОС |
| 0 | Сетевые ОС |
| 0 | Все варианты не верны |

Вопрос №56

|  |  |
| --- | --- |
|  | **Классы прикладного ПО?** |
| 1 | Все перечисленное |
| 0 | Общего назначения |
| 0 | Проблемно-ориентированные |
| 0 | Информационное поисковые |
| 0 | Мультимедиа |

Вопрос №57

|  |  |
| --- | --- |
|  | **Виды системного ПО?** |
| 1 | Базовое ПО, Сервисное ПО |
| 0 | Вредоносное ПО, Базовое ПО |
| 0 | Сервисное ПО, Вредоносное ПО |
| 0 | Вредоносное ПО, Прикладное ПО |
| 0 | Нет правильного ответа |

Вопрос №58

|  |  |
| --- | --- |
|  | **При выполнении параллельного запроса порядок данных в результирующей выборки может быть?** |
| 1 | Не предсказуем |
| 0 | Предсказуем |
| 0 | Не предсказуемо предсказуем |
| 0 | Оба варианта верны |
| 0 | Все варианты верны |

Вопрос №59

|  |  |
| --- | --- |
|  | **Начиная с какой версии NET в пространство имен System.Linq был добавлен класс ParallelEnumerable?** |
| 1 | 4.0 |
| 0 | 4.3 |
| 0 | 3.0 |
| 0 | 3.8 |
| 0 | 3.5 |

Вопрос №60

|  |  |
| --- | --- |
|  | **Какой метод позволяет распараллелить запрос к источнику данных?** |
| 1 | AsParallel() |
| 0 | ForAll |
| 0 | SayHello |
| 0 | SayGoodbye |
| 0 | Все верны |

Вопрос №61

|  |  |
| --- | --- |
|  | **Инструментарий технологии программирования – это** |
| 1 | Программные продукты, предназначенные для поддержки технологии программирования |
| 0 | Универсальное устройство для передачи информации |
| 0 | Устройство для удаления информации |
| 0 | Программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем. |
| 0 | Все ответы верны |

Вопрос №62

|  |  |
| --- | --- |
|  | **CASE-технология - это** |
| 1 | Программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем. |
| 0 | Программный продукт, предназначенный для поддержки технологии программирования |
| 0 | Устройство для удаления информации |
| 0 | Универсальное устройство для передачи информации |
| 0 | Все ответы верны |

Вопрос №63

|  |  |
| --- | --- |
|  | **Средства CASE-технологий бывают** |
| 1 | Встроенные и независимые от системы реализации |
| 0 | Встроенные в систему реализации |
| 0 | Зависимые от системы реализации |
| 0 | Независимые от системы реализации |
| 0 | Все ответы верны |

Вопрос №64

|  |  |
| --- | --- |
|  | **Инструментальная среда пользователя - это** |
| 1 | Специальные средства, встроенные в пакеты прикладных программ |
| 0 | Универсальное устройство для передачи информации |
| 0 | Программные продукты, предназначенные для поддержки технологии программирования |
| 0 | Программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем. |
| 0 | Все ответы верны |

Вопрос №65

|  |  |
| --- | --- |
|  | **Отладчики – это** |
| 1 | Специальные программы, предназначенные для трассировки и анализа выполнения других программ. |
| 0 | Специальные средства, встроенные в пакеты прикладных программ |
| 0 | Устройство для удаления информации |
| 0 | Программные продукты, предназначенные для поддержки технологии программирования |
| 0 | Все ответы верны |

Вопрос №66

|  |  |
| --- | --- |
|  | **Системы программирования включают:** |
| 1 | Все ответы верны |
| 0 | отладчик |
| 0 | набор библиотек |
| 0 | редактор связей |
| 0 | компилятор (транслятор) |

Вопрос №67

|  |  |
| --- | --- |
|  | **Язык программирования - это** |
| 1 | Формализованный язык для описания алгоритма решения задач на компьютере |
| 0 | Специальные программы, предназначенные для трассировки и анализа выполнения других программ. |
| 0 | Программные продукты, предназначенные для поддержки технологии программирования |
| 0 | Программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем. |
| 0 | Все ответы верны |

Вопрос №68

|  |  |
| --- | --- |
|  | **Классы языков программирования:** |
| 1 | Все ответы верны |
| 0 | машинные языки |
| 0 | машинно-ориентированные языки |
| 0 | процедурно-ориентированные языки |
| 0 | проблемно-ориентированные языки |

Вопрос №69

|  |  |
| --- | --- |
|  | **Модульное программирование предполагает:** |
| 1 | разбиение программы на независимые модули |
| 0 | разработку программного обеспечения, основанную на объектах |
| 0 | В программе базовые конструкции могут быть вложены друг в друга произвольным образом, но никаких других средств управления последовательностью выполнения операций не предусматривается. |
| 0 | Нет верного ответа |
| 0 | Все ответы верны |

Вопрос №70

|  |  |
| --- | --- |
|  | **Объектно-ориентированное программирование – это** |
| 1 | подход к разработке программного обеспечения, основанный на объектах, а не на процедурах. |
| 0 | Программные продукты, предназначенные для поддержки технологии программирования |
| 0 | разбиение программы на независимые модули |
| 0 | Нет верного ответа |
| 0 | Все ответы верны |

Вопрос №71

|  |  |
| --- | --- |
|  | **Программный продукт --** |
| 1 | комплекс взаимосвязанных программ для решения определенной проблемы (задачи) массового спроса, подготовленный к реализации как любой вид промышленной продукции. |
| 0 | специальные средства, встроенные в пакеты прикладных программ |
| 0 | разбиение программы на независимые модули |
| 0 | специальные средства, встроенные в пакеты прикладных программ |
| 0 | Все ответы верны |

Вопрос №72

|  |  |
| --- | --- |
|  | **Этапы разработки программного продукта** |
| 1 | Постановка задачи->Анализ и исследование задачи, построение модели->Разработка алгоритма->Программирование-> Тестирование и отладка-> Анализ результатов решения-> Сопровождение программы |
| 0 | Сопровождение программы -> Постановка задачи-> Программирование-> Тестирование и отладка-> Анализ результатов решения-> Анализ и исследование задачи, построение модели->Разработка алгоритма |
| 0 | Сопровождение программы -> Программирование-> Постановка задачи-> Тестирование и отладка-> Анализ и исследование задачи, построение модели-> Анализ результатов решения->Разработка алгоритма |
| 0 | Программирование-> построение модели->Сопровождение программы -> Постановка задачи-> Тестирование и отладка-> Анализ и исследование задачи, Анализ результатов решения->Разработка алгоритма |
| 0 | Программирование-> Тестирование и отладка-> построение модели->Сопровождение программы -> Постановка задачи-> Анализ и исследование задачи, Анализ результатов решения->Разработка алгоритма |

Вопрос №73

|  |  |
| --- | --- |
|  | **Сопровождение программного продукта - это** |
| 1 | поддержка работоспособности программного продукта, переход на его новые версии, внесение изменений, исправление обнаруженных ошибок |
| 0 | комплекс взаимосвязанных программ для решения определенной проблемы (задачи) массового спроса, подготовленный к реализации как любой вид промышленной продукции. |
| 0 | Программные продукты, предназначенные для поддержки технологии программирования |
| 0 | Нет верного ответа |
| 0 | Все ответы верны |

Вопрос №74

|  |  |
| --- | --- |
|  | **На создание программных продуктов затрачиваются значительные ресурсы:** |
| 1 | Все варианты верны |
| 0 | материальные |
| 0 | финансовые |
| 0 | Нет верного ответа |
| 0 | трудовые |

Вопрос №75

|  |  |
| --- | --- |
|  | **Постановка задачи– это** |
| 1 | точная формулировка условий задачи с описанием входной и выходной информации |
| 0 | особый метод для создания математического способа решения проблемы |
| 0 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | испытание, проверка правильности работы программы в целом, либо её составных частей |
| 0 | процесс создания компьютерных программ |

Вопрос №76

|  |  |
| --- | --- |
|  | **Разработка алгоритма – это** |
| 1 | особый метод для создания математического способа решения проблемы |
| 0 | точная формулировка условий задачи с описанием входной и выходной информации |
| 0 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | испытание, проверка правильности работы программы в целом, либо её составных частей |
| 0 | процесс создания компьютерных программ |

Вопрос №77

|  |  |
| --- | --- |
|  | **Отладка программы – это** |
| 1 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | точная формулировка условий задачи с описанием входной и выходной информации |
| 0 | особый метод для создания математического способа решения проблемы |
| 0 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | испытание, проверка правильности работы программы в целом, либо её составных частей |

Вопрос №78

|  |  |
| --- | --- |
|  | **Тестирование программы – это** |
| 1 | испытание, проверка правильности работы программы в целом, либо её составных частей |
| 0 | точная формулировка условий задачи с описанием входной и выходной информации |
| 0 | особый метод для создания математического способа решения проблемы |
| 0 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | испытание, проверка правильности работы программы в целом, либо её составных частей |

Вопрос №79

|  |  |
| --- | --- |
|  | **Программирование – это** |
| 1 | процесс создания компьютерных программ |
| 0 | точная формулировка условий задачи с описанием входной и выходной информации |
| 0 | особый метод для создания математического способа решения проблемы |
| 0 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | испытание, проверка правильности работы программы в целом, либо её составных частей |

Вопрос №80

|  |  |
| --- | --- |
|  | **Машинный язык – это** |
| 1 | система инструкций и данных, непосредственно выполняемых центральным процессором в компьютере |
| 0 | точная формулировка условий задачи с описанием входной и выходной информации |
| 0 | особый метод для создания математического способа решения проблемы |
| 0 | это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере |
| 0 | испытание, проверка правильности работы программы в целом, либо её составных частей |

Вопрос №81

|  |  |
| --- | --- |
|  | **Укажите что относится к «водопадной» модели ЖЦП** |
| 1 | Невозможность вернуться на предыдущюю фазу проектирования |
| 0 | Модель не требует определить опорные точки, в которых будет оцениваться сделанное и решаться вопрос о том, можно ли двигаться дальше |
| 0 | Возможность вернуться на предыдущюю фазу проектирования |
| 0 | Модель применяется для сложных проектов, когда требования могут неоднократно меняться |
| 0 | Все ответы верные |

Вопрос №82

|  |  |
| --- | --- |
|  | **Укажите что относится к «водопадной» модели ЖЦП** |
| 1 | Модель требует определить опорные точки, в которых будет оцениваться сделанное и решаться вопрос о том, можно ли двигаться дальше |
| 0 | Отсутствует необходимость готовить огромную массу документации и не требует единообразной процедуры оценки результатов на каждом этапе |
| 0 | Возможность вернуться на предыдущюю фазу проектирования |
| 0 | Модель применяется для сложных проектов, когда требования могут неоднократно меняться |
| 0 | Все ответы верные |

Вопрос №83

|  |  |
| --- | --- |
|  | **Укажите что относится к «водопадной» модели ЖЦП** |
| 1 | модель хороша для проектов, в которых требования легко формулируются с самого начала |
| 0 | Отсутствует необходимость готовить огромную массу документации и не требует единообразной процедуры оценки результатов на каждом этапе |
| 0 | Возможность вернуться на предыдущюю фазу проектирования |
| 0 | Модель применяется для сложных проектов, когда требования могут неоднократно меняться |
| 0 | Все ответы верные |

Вопрос №84

|  |  |
| --- | --- |
|  | **Укажите что относится к «водопадной» модели ЖЦП** |
| 1 | Модель вынуждает готовить огромную массу документации и требует единообразной процедуры оценки результатов на каждом этапе |
| 0 | Отсутствует необходимость готовить огромную массу документации и не требует единообразной процедуры оценки результатов на каждом этапе |
| 0 | Возможность вернуться на предыдущюю фазу проектирования |
| 0 | Модель применяется для сложных проектов, когда требования могут неоднократно меняться |
| 0 | Все ответы верные |

Вопрос №85

|  |  |
| --- | --- |
|  | **Укажите что относится к «водопадной» модели ЖЦП** |
| 1 | Все ответы верные |
| 0 | Модель вынуждает готовить огромную массу документации и требует единообразной процедуры оценки результатов на каждом этапе |
| 0 | Отсутствует возможность вернуться на предыдущюю фазу проектирования |
| 0 | Модель не применяется для сложных проектов, когда требования могут неоднократно меняться |
| 0 | Модель требует определить опорные точки, в которых будет оцениваться сделанное и решаться вопрос о том, можно ли двигаться дальше |

Вопрос №86

|  |  |
| --- | --- |
|  | **Укажите что относится к модели создания прототипов** |
| 1 | Все ответы верные |
| 0 | С самого начала разработчики пытаются выделить основные, существенные требования заказчика и реализовать только их в виде работающего прототипа системы |
| 0 | Цикл разработки и показа прототипа повторяется несколько раз, пока заказчик не скажет: «Да, это, кажется, то, что мне нужно» |
| 0 | Одна из первых практически полезных моделей ЖЦП |
| 0 | компоненты, выброшенные в начале разработки доробатываются после разработки и утверждения основых требований заказчика |

Вопрос №87

|  |  |
| --- | --- |
|  | **Укажите что относится к спиральной модели** |
| 1 | Разработка приложения выглядит как серия последовательных итераций |
| 0 | На первых этапах не уточняются спецификации продукта |
| 0 | На первых этапах только добавляются новые возможности и функции |
| 0 | На первых этапах осуществляют оценку рисков продолжения работ |
| 0 | Все ответы верные |

Вопрос №88

|  |  |
| --- | --- |
|  | **Укажите что относится к спиральной модели** |
| 1 | Все ответы верные |
| 0 | На первых этапах уточняются спецификации продукта |
| 0 | На последующих этапах добавляются новые возможности и функции |
| 0 | По окончании каждой этерации осуществляют оценку рисков продолжения работ |
| 0 | Разработка приложения выглядит как серия последовательных итераций |

Вопрос №89

|  |  |
| --- | --- |
|  | **Укажите что относится к спиральной модели** |
| 1 | Спиральная модель допускает корректировки по ходу работы |
| 0 | Спиральная модель не допускает корректировки по ходу работы |
| 0 | При большом числе этераций разработка по этой модели не нуждается в глубокой автоматизации всех процессов |
| 0 | Корректировки апо ходу работы не способствуют улучшению продукта |
| 0 | Все ответы верные |

Вопрос №9

|  |  |
| --- | --- |
|  | **Укажите как формулирует «системные требования» К.Вигерс** |
| 1 | Наиболее важные, существенные требования, которые относятся в целом к системе и не содержат избыточной детализации |
| 0 | Требования , которые относятся к системе в целом и содержат избыточную детализацию |
| 0 | Детальное определение системы |
| 0 | Требования к программному обеспечению |
| 0 | Все ответы верные |

Вопрос №91

|  |  |
| --- | --- |
|  | **Укажите как формулирует «системные требования» INCOSE** |
| 1 | требования, выдвигаемые прикладной программной системой (в частности - информационной) к среде своего функционирования (системной, аппаратной). |
| 0 | Требования , которые относятся к системе в целом и не содержат избыточную детализацию |
| 0 | Детальное определение системы |
| 0 | Наиболее важные, существенные требования, которые относятся в целом к системе и не содержат избыточной детализации |
| 0 | Все ответы верные |

Вопрос №92

|  |  |
| --- | --- |
|  | **Укажите что относится к функциональным требованиям продукта** |
| 1 | Все ответы верные |
| 0 | Регламентируют функционирование и поведение ситсемы |
| 0 | Отвечают на вопрос «Что должна делать система» в различных ситуациях |
| 0 | Определяют основной «фронт работ» Разработчика |
| 0 | Устанавливают цели, задачи и сервис, предоставляемые системой Заказчику |

Вопрос №93

|  |  |
| --- | --- |
|  | **Укажите основные понятия связанные с вариантом использования (Use case)** |
| 1 | Акторы, прецеденты и объект |
| 0 | Объект, субъект, предикат |
| 0 | Актор, субъект, объект |
| 0 | Прецеденты, субъекты, акторы |
| 0 | Все ответы верные |

Вопрос №94

|  |  |
| --- | --- |
|  | **Укажите что относится к нефункциональным требованиям** |
| 1 | Регламентируют внутренние и внешние условия или атрибуты функционирования системы |
| 0 | Отвечают на вопрос «Что должна делать система» |
| 0 | Устанавливают цели, задачи и сервисы, представляемые системой Заказчику |
| 0 | Как правило записываются предписывающими правилами |
| 0 | Все ответы верные |

Вопрос №95

|  |  |
| --- | --- |
|  | **Укажите группы нефункциональных требований выделяемых К. Вигерсом** |
| 1 | Внешние интерфейсы, Атрибуты качества, Ограничения |
| 0 | Внутриние интерфейсы, Атрибуты качества, Ограничения |
| 0 | Атрибуты качества, Атрибуты количества |
| 0 | Ограничения, предписания |
| 0 | Все ответы верные |

Вопрос №96

|  |  |
| --- | --- |
|  | **Укажите оснвные атрибуты качества** |
| 1 | Все ответы верные |
| 0 | Надежность, производительность |
| 0 | Производительность,применимость |
| 0 | Эксплуатационная пригодность, надежность |
| 0 | Применимость, производительность |

Вопрос №96

|  |  |
| --- | --- |
|  | **Укажите оснвные атрибуты качества** |
| 1 | Все ответы верные |
| 0 | Надежность, производительность |
| 0 | Производительность,применимость |
| 0 | Эксплуатационная пригодность, надежность |
| 0 | Применимость, производительность |

Вопрос №97

|  |  |
| --- | --- |
|  | **Укажите критерий хорошей архитектуры имеющий следующее определение:**  **Решать поставленные задачи и хорошо выполнять свои функции, причем в различных условиях.** |
| 1 | Эффективность системы |
| 0 | Гибкость системы |
| 0 | Расширяемость системы |
| 0 | Масштабируемость системы |
| 0 | Тестируемост системы |

Вопрос №98

|  |  |
| --- | --- |
|  | **Укажите критерий хорошей архитектуры имеющий следующее определение:**  **Быстрота и удобство внесения изменений в существующий функционал** |
| 1 | Гибкость системы |
| 0 | Эффективность системы |
| 0 | Расширяемость системы |
| 0 | Масштабируемость системы |
| 0 | Тестируемост системы |

Вопрос №99

|  |  |
| --- | --- |
|  | **Укажите критерий хорошей архитектуры, имеющий следующее определение:**  **Возможность добавлять в систему новые сущности и функции, не нарушая ее основной структуры.** |
| 1 | Расширяемость системы |
| 0 | Эффективность системы |
| 0 | Гибкость системы |
| 0 | Масштабируемость системы |
| 0 | Тестируемост системы |

Вопрос №100

|  |  |
| --- | --- |
|  | **Укажите критерий хорошей архитектуры, имеющий следующее определение:**  **Возможность сократить срок разработки за счёт добавления к проекту новых людей** |
| 1 | Масштабируемость системы |
| 0 | Эффективность системы |
| 0 | Гибкость системы |
| 0 | Расширяемость системы |
| 0 | Тестируемост системы |

Вопрос №101

|  |  |
| --- | --- |
|  | **Функционально прочный модуль - это** |
| 1 | модуль, выполняющий (реализующий) одну какую-либо определенную функцию |
| 0 | модуль, имеющий определённое свойство при работе с определенной функцией |
| 0 | модуль, выполняющий (реализующий) несколько операций (функций) над одной и той же структурой данных (информационным объектом), которая считается неизвестной вне этого модуля. |
| 0 | модуль который управляет выполнением нижестоящих функций, которым соответствуют подчиненные модули |
| 0 | модуль, выполняющий (реализующий) несколько разных функций |

Вопрос №12

|  |  |
| --- | --- |
|  | **Модуль это** |
| 1 | функционально законченный фрагмент программы |
| 0 | Элемент определяющий работу всей программы |
| 0 | Весь программный код приложения |
| 0 | Минимальная единица элемента кода программы |
| 0 | Основная функция программы |

Вопрос №103

|  |  |
| --- | --- |
|  | **Прочность модуля это** |
| 1 | мера его внутренних связей |
| 0 | информационная наполненность модуля |
| 0 | способность модуля выполнять определенные функции |
| 0 | ресурс модуля |
| 0 | Функциональность модуля |

Вопрос №104

|  |  |
| --- | --- |
|  | **Информационно прочный модуль - это** |
| 1 | модуль, выполняющий (реализующий) несколько операций (функций) над одной и той же структурой данных (информационным объектом), которая считается неизвестной вне этого модуля |
| 0 | модуль, выполняющий (реализующий) одну какую-либо определенную функцию |
| 0 | модуль, имеющий определённое свойство при работе с определенной функцией |
| 0 | модуль который управляет выполнением нижестоящих функций, которым соответствуют подчиненные модули |
| 0 | модуль, выполняющий (реализующий) несколько разных функций |

Вопрос №105

|  |  |
| --- | --- |
|  | **Сцепление модуля это** |
| 1 | мера его зависимости по данным от других модулей |
| 0 | мера его внутренних связей |
| 0 | информационная наполненность модуля |
| 0 | способность модуля выполнять определенные функции |
| 0 | ресурс модуля |

Вопрос №106

|  |  |
| --- | --- |
|  | **Рутинность модуля это** |
| 1 | его независимость от предыстории обращений к нему |
| 0 | мера его зависимости по данным от других модулей |
| 0 | мера его внутренних связей |
| 0 | информационная наполненность модуля |
| 0 | способность модуля выполнять определенные функции |

Вопрос №107

|  |  |
| --- | --- |
|  | **При определении набора модулей, реализующих функции конкретного алгоритма, необходимо учитывать** |
| 1 | Все верны |
| 0 | принятие основных решений в алгоритме выносится на максимально "высокий" по иерархии уровень; |
| 0 | для использования одной и той же функции в разных местах алгоритма создается один модуль, который вызывается на выполнение по мере необходимости |
| 0 | В результате дальнейшей детализации алгоритма создается функционально-модульная схема (ФМС) алгоритма приложения, которая является основой для программирования |
| 0 | каждый модуль вызывается на выполнение вышестоящим модулем и, закончив работу, возвращает управление вызвавшему его модулю; |

Вопрос №108

|  |  |
| --- | --- |
|  | **Единственным видом сцепления модулей, который рекомендуется для использования современной технологией программирования, является** |
| 1 | параметрическое сцепление |
| 0 | сцепление по образцу |
| 0 | сцепление по общей области |
| 0 | сцепление по управлению |
| 0 | сцепление по внешним ссылкам |

Вопрос №109

|  |  |
| --- | --- |
|  | **применительно к средствам СУБД отдельными модулями могут быть:** |
| 1 | Все верны |
| 0 | экранные формы ввода и/или редактирования информации базы данных; |
| 0 | отчеты генератора отчетов; |
| 0 | макросы; |
| 0 | стандартные процедуры обработки информации; |

Вопрос №110

|  |  |
| --- | --- |
|  | **Худшим видом сцепления модулей является** |
| 1 | сцепление по содержимому |
| 0 | сцепление по общей области |
| 0 | сцепление по управлению |
| 0 | сцепление по данным |
| 0 | Смешанное сцепление |

Вопрос №111

|  |  |
| --- | --- |
|  | **Модуль характеризуют** |
| 1 | Все верны |
| 0 | один вход и один выход - на входе программный модуль получает определенный набор исходных данных, выполняет содержательную обработку и возвращает один набор результатных данных, т.е. реализуется стандартный принцип IPO (Input - Process - Output) - вход-процесс-выход; |
| 0 | функциональная завершенность - модуль выполняет перечень регламентированных операций для реализации каждой отдельной функции в полном составе, достаточных для завершения начатой обработки; |
| 0 | логическая независимость - результат работы программного модуля зависит только от исходных данных, но не зависит от работы других модулей; |
| 0 | слабые информационные связи с другими программными модулями - обмен информацией между модулями должен быть по возможности минимизирован; |

Вопрос №112

|  |  |
| --- | --- |
|  | **Модуль характеризуют** |
| 1 | Все верны |
| 0 | один вход и один выход - на входе программный модуль получает определенный набор исходных данных, выполняет содержательную обработку и возвращает один набор результатных данных, т.е. реализуется стандартный принцип IPO (Input - Process - Output) - вход-процесс-выход; |
| 0 | функциональная завершенность - модуль выполняет перечень регламентированных операций для реализации каждой отдельной функции в полном составе, достаточных для завершения начатой обработки; |
| 0 | логическая независимость - результат работы программного модуля зависит только от исходных данных, но не зависит от работы других модулей; |
| 0 | слабые информационные связи с другими программными модулями - обмен информацией между модулями должен быть по возможности минимизирован; |

Вопрос №113

|  |  |
| --- | --- |
|  | **Сцепление по общей области** |
| 1 | Тип зацепления, при котором два программных модуля совместно используют общую область данных. |
| 0 | Тип зацепления, при котором некоторые или все программные модули включены в некоторый модуль как составные части. |
| 0 | Тип зацепления, при котором один программный модуль обменивается данными с другим модулем с явной целью повлиять на его последующее выполнение. |
| 0 | Тип зацепления, при котором выходные данные одного программного модуля служат входными данными другого модуля. |
| 0 | Тип зацепления, при котором различные подмножества значений некоторого элемента данных используются в нескольких программных модулях для разных и несвязанных целей. |

Вопрос №114

|  |  |
| --- | --- |
|  | **цепление по содержимому** |
| 1 | Тип зацепления, при котором некоторые или все программные модули включены в некоторый модуль как составные части. |
| 0 | Тип зацепления, при котором два программных модуля совместно используют общую область данных. |
| 0 | Тип зацепления, при котором один программный модуль обменивается данными с другим модулем с явной целью повлиять на его последующее выполнение. |
| 0 | Тип зацепления, при котором выходные данные одного программного модуля служат входными данными другого модуля. |
| 0 | Тип зацепления, при котором различные подмножества значений некоторого элемента данных используются в нескольких программных модулях для разных и несвязанных целей. |

Вопрос №115

|  |  |
| --- | --- |
|  | **Зацепление по управлению** |
| 1 | Тип зацепления, при котором один программный модуль обменивается данными с другим модулем с явной целью повлиять на его последующее выполнение. |
| 0 | Тип зацепления, при котором два программных модуля совместно используют общую область данных. |
| 0 | Тип зацепления, при котором некоторые или все программные модули включены в некоторый модуль как составные части. |
| 0 | Тип зацепления, при котором выходные данные одного программного модуля служат входными данными другого модуля. |
| 0 | Тип зацепления, при котором различные подмножества значений некоторого элемента данных используются в нескольких программных модулях для разных и несвязанных целей. |

Вопрос №116

|  |  |
| --- | --- |
|  | **Зацепление по данным** |
| 1 | Тип зацепления, при котором выходные данные одного программного модуля служат входными данными другого модуля. |
| 0 | Тип зацепления, при котором два программных модуля совместно используют общую область данных. |
| 0 | Тип зацепления, при котором некоторые или все программные модули включены в некоторый модуль как составные части. |
| 0 | Тип зацепления, при котором один программный модуль обменивается данными с другим модулем с явной целью повлиять на его последующее выполнение. |
| 0 | Тип зацепления, при котором различные подмножества значений некоторого элемента данных используются в нескольких программных модулях для разных и несвязанных целей. |

Вопрос №117

|  |  |
| --- | --- |
|  | **Смешанное зацепление** |
| 1 | Тип зацепления, при котором различные подмножества значений некоторого элемента данных используются в нескольких программных модулях для разных и несвязанных целей. |
| 0 | Тип зацепления, при котором два программных модуля совместно используют общую область данных. |
| 0 | Тип зацепления, при котором некоторые или все программные модули включены в некоторый модуль как составные части. |
| 0 | Тип зацепления, при котором один программный модуль обменивается данными с другим модулем с явной целью повлиять на его последующее выполнение. |
| 0 | Тип зацепления, при котором выходные данные одного программного модуля служат входными данными другого модуля. |

Вопрос №118

|  |  |
| --- | --- |
|  | **Патологическое зацепление** |
| 1 | Тип зацепления, при котором один программный модуль зависит от деталей внутренней реализации другого модуля или влияет на них. |
| 0 | Тип зацепления, при котором два программных модуля совместно используют общую область данных. |
| 0 | Тип зацепления, при котором некоторые или все программные модули включены в некоторый модуль как составные части. |
| 0 | Тип зацепления, при котором один программный модуль обменивается данными с другим модулем с явной целью повлиять на его последующее выполнение. |
| 0 | Тип зацепления, при котором выходные данные одного программного модуля служат входными данными другого модуля. |

Вопрос №119

|  |  |
| --- | --- |
|  | **Роль модулей могут играть** |
| 1 | Все верны |
| 0 | структуры данных |
| 0 | библиотеки функций |
| 0 | классы |
| 0 | сервисы |

Вопрос №120

|  |  |
| --- | --- |
|  | **модули содержат** |
| 1 | Все верны |
| 0 | определение доступных для обработки данных |
| 0 | операции обработки данных |
| 0 | схемы взаимосвязи с другими модулями |
| 0 | Спецификацию и тело |

Вопрос №121

|  |  |
| --- | --- |
|  | **Какие компоненты могут быть структурными для программной системы или программного комплекса** |
| 1 | Все перечисленные |
| 0 | Программы |
| 0 | БД |
| 0 | Подсистемы |
| 0 | Библиотеки ресурсов |

Вопрос №122

|  |  |
| --- | --- |
|  | **Какими достоинствами обладают case-технологии?** |
| 1 | Все нижеперечисленное |
| 0 | улучшают качество создаваемого ПО за счет средств автоматического контроля |
| 0 | ускоряют процесс проектирования и разработки; |
| 0 | позволяют разработчику больше времени уделять творческой работе по созданию ПО |
| 0 | поддерживают развитие и сопровождение разработки |

Вопрос №123

|  |  |
| --- | --- |
|  | **Укажите преимущества спиральной модели, на которой базируется CASE-технология** |
| 1 | Все нижеперечисленное |
| 0 | накопление и повторное использование программных средств, моделей и прототипов |
| 0 | ориентация на развитие и модификацию ПО в процессе проектирования |
| 0 | анализ риска и издержек в процессе проектирования |
| 0 | Нет верного ответа |

Вопрос №124

|  |  |
| --- | --- |
|  | **Укажите что на основе некоторого подхода определяет шаги работы, их последовательность, а также правила распределения и назначения методов?** |
| 1 | методология |
| 0 | метод |
| 0 | нотация |
| 0 | средства |
| 0 | Нет верного ответа |

Вопрос №125

|  |  |
| --- | --- |
|  | **Укажите что определяет способ достижения той или иной цели?** |
| 1 | метод |
| 0 | методология |
| 0 | нотация |
| 0 | средства |
| 0 | Нет верного ответа |

Вопрос №126

|  |  |
| --- | --- |
|  | **Укажите что называют системой обозначений, используемых для описания структуры системы, элементов данных, этапов обработки и других компонентов?** |
| 1 | нотация |
| 0 | методология |
| 0 | метод |
| 0 | средства |
| 0 | нет верного ответа |

Вопрос №127

|  |  |
| --- | --- |
|  | **инструментарий для поддержки методов это?** |
| 1 | средства |
| 0 | нотация |
| 0 | методология |
| 0 | Структуры |
| 0 | Все ответы верны |

Вопрос №128

|  |  |
| --- | --- |
|  | **Какие факторы способствовали появлению CASE-технологии?** |
| 1 | Все варианты верны |
| 0 | Нет верного ответа |
| 0 | подготовка аналитиков и программистов, восприимчивых к концепциям модульного и структурного программирования |
| 0 | широкое внедрение и постоянный рост производительности компьютеров |
| 0 | внедрение сетевой технологии |

Вопрос №129

|  |  |
| --- | --- |
|  | **Что является средством для реализации структурного анализа и построения структуры конкретной системы?** |
| 1 | структуры |
| 0 | средства |
| 0 | нотация |
| 0 | методология |
| 0 | метод |

Вопрос №130

|  |  |
| --- | --- |
|  | **Укажите факторы, усложняющие определение возможного эффекта от использования CASE-средств** |
| 1 | Все нижеперечисленные |
| 0 | широкое разнообразие качества и возможностей CASE-средств |
| 0 | широкое разнообразие в практике внедрения различных организаций |
| 0 | широкий диапазон предметных областей проектов |
| 0 | различная степень интеграции CASE-средств в различных проектах |

Вопрос №131

|  |  |
| --- | --- |
|  | **Укажите программное средство, выполняющее моделирование бизнес-процессов?** |
| 1 | BPwin |
| 0 | ERwin |
| 0 | ERwin Examiner |
| 0 | ModelMart |
| 0 | Rational Rose |

Вопрос №132

|  |  |
| --- | --- |
|  | **Укажите программное средство, выполняющее моделирование баз данных и хранилищ данных?** |
| 1 | ERwin |
| 0 | ERwin Examiner |
| 0 | ModelMart |
| 0 | Rational Rose |
| 0 | BPwin |

Вопрос №133

|  |  |
| --- | --- |
|  | **Укажите программное средство, выполняющее проверку структуры СУБД и моделей, созданных в Erwin** |
| 1 | ERwin Examiner |
| 0 | ModelMart |
| 0 | Rational Rose |
| 0 | BPwin |
| 0 | ModelMart |

Вопрос №134

|  |  |
| --- | --- |
|  | **Укажите среду для командной работы проектировщиков?** |
| 1 | ModelMart |
| 0 | Rational Rose |
| 0 | BPwin |
| 0 | ERwin Examiner |
| 0 | Paradigm Plus |

Вопрос №135

|  |  |
| --- | --- |
|  | **Укажите программное средство, выполняющее моделирование бизнес-процессов и компонентов приложений?** |
| 1 | Rational Rose |
| 0 | BPwin |
| 0 | ERwin Examiner |
| 0 | Paradigm Plus |
| 0 | ModelMart |

Вопрос №136

|  |  |
| --- | --- |
|  | **Укажите программное средство, выполняющее моделирование приложений и генерация объектного кода?** |
| 1 | Paradigm Plus |
| 0 | Rational Rose |
| 0 | BPwin |
| 0 | ERwin Examiner |
| 0 | ModelMart |

Вопрос №137

|  |  |
| --- | --- |
|  | **Укажите высокофункциональное средство проектирования программных систем и баз данных, реализующее технологию CASE и собственную методологию Oracle – CDM?** |
| 1 | Oracle Designer |
| 0 | Paradigm Plus |
| 0 | Rational Rose |
| 0 | BPwin |
| 0 | ERwin Examiner |

Вопрос №138

|  |  |
| --- | --- |
|  | **Укажите программное средство, позволяющее команде разработчиков полностью провести проект, начиная от анализа бизнес-процессов через моделирование к генерации кода и получению прототипа, а в дальнейшем и окончательного продукта?** |
| 1 | Oracle Designer |
| 0 | Paradigm Plus |
| 0 | Rational Rose |
| 0 | BPwin |
| 0 | ERwin Examiner |

Вопрос №139

|  |  |
| --- | --- |
|  | **Для успешного внедрения CASE-средств организация должна обладать какими качествами?** |
| 1 | Все перечисленные |
| 0 | Нет верного ответа |
| 0 | Культура |
| 0 | Управление |
| 0 | Технология |

Вопрос №140

|  |  |
| --- | --- |
|  | **Под каким термином понимаются программные средства, поддерживающие процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного программного обеспечения и баз данных?** |
| 1 | "CASE-средства" |
| 0 | Paradigm Plus |
| 0 | Rational Rose |
| 0 | BPwin |
| 0 | Нет верного ответа |

Вопрос №141

|  |  |
| --- | --- |
|  | **Структурной называют схему, которая** |
| 1 | Отражает состав и взаимодействие ПО |
| 0 | показывает отчеты |
| 0 | отражает макросы |
| 0 | видит экранную форму |
| 0 | ищет многозадачность |

Вопрос №142

|  |  |
| --- | --- |
|  | **Самый простой вид ПО - это?** |
| 1 | Программа |
| 0 | Макрос |
| 0 | Технология |
| 0 | Часть |
| 0 | Вид |

Вопрос №143

|  |  |
| --- | --- |
|  | **С помощью какого метода выполняется разработка структурной схемы?** |
| 1 | Пошаговая детализация |
| 0 | Матрица |
| 0 | Пиксель |
| 0 | Подпрограмма |
| 0 | Библиотека |

Вопрос №144

|  |  |
| --- | --- |
|  | **Как представлен результат уточнения структуры?** |
| 1 | В виде структурной и/или функциональной схемы |
| 0 | В виде матрицы |
| 0 | В виде незаполненной таблицы |
| 0 | В виде макроса |
| 0 | В виде презентации |

Вопрос №145

|  |  |
| --- | --- |
|  | **Какая схема даёт более полное представление о проектируемом ПО?** |
| 1 | Функциональная |
| 0 | Подпрограмма |
| 0 | Программа |
| 0 | Матричная |
| 0 | Пиксельная |

Вопрос №146

|  |  |
| --- | --- |
|  | **Что используют для изображения функциональных схем?** |
| 1 | Специальные изображения |
| 0 | Файлы |
| 0 | Устройства |
| 0 | Пиксели |
| 0 | Карты |

Вопрос №147

|  |  |
| --- | --- |
|  | **При помощи какого подхода осуществляется декомпозиция программ?** |
| 1 | Структурный подход |
| 0 | математический подход |
| 0 | Пиксельный подход |
| 0 | Многоуровневый подход |
| 0 | Физический подход |

Вопрос №148

|  |  |
| --- | --- |
|  | **Результат декомпозиции структурного подхода** |
| 1 | Структурная схема |
| 0 | Физическая схема |
| 0 | Пиксель |
| 0 | Таблица |
| 0 | Задача |

Вопрос №149

|  |  |
| --- | --- |
|  | **С помощью чего определяются состав и вид программных модулей?** |
| 1 | С помощью инструментальных средств |
| 0 | С помощью макросов |
| 0 | С помощью таблиц |
| 0 | С помощью презентации |
| 0 | С помощью картинки |

Вопрос №150

|  |  |
| --- | --- |
|  | **Разница между функциональными и структурными схемами?** |
| 1 | Более информативны |
| 0 | Более медленные |
| 0 | Показывают количество строк |
| 0 | показывают количество пикселей |
| 0 | Показывают количество знаков |

Вопрос №151

|  |  |
| --- | --- |
|  | **Сколько уровней иерархии минимально отображает структурная схема программы?** |
| 1 | 2 уровня |
| 0 | 3 уровня |
| 0 | 7 уровней |
| 0 | 10 уровней |
| 0 | 1 уровень |

Вопрос №152

|  |  |
| --- | --- |
|  | **На чем базируется метод пошаговой детализации?** |
| 1 | На структурном программировании |
| 0 | На алгоритме |
| 0 | На физическом уровне |
| 0 | На детализации |
| 0 | На картинке |

Вопрос №153

|  |  |
| --- | --- |
|  | **С чего начинают процесс проектирования сложного ПО?** |
| 1 | С уточнения структуры |
| 0 | С картинки |
| 0 | С текста |
| 0 | С детализации |
| 0 | С пикселя |

Вопрос №154

|  |  |
| --- | --- |
|  | **Какой ГОСТ имеет функциональны схема?** |
| 1 | 19.701-90 |
| 0 | 18.554-1 |
| 0 | 126-76 |
| 0 | 274.3-7 |
| 0 | 777-3 |

Вопрос №195

|  |  |
| --- | --- |
|  | **Сколько уровней иерархии минимально отображает структурная схема?** |
| 1 | 2 уровня |
| 0 | 1 уровень |
| 0 | 5 уровней |
| 0 | 8 уровней |
| 0 | Ни одного уровня |

Вопрос №156

|  |  |
| --- | --- |
|  | **Что такое информационно прочный модуль?** |
| 1 | Модуль, который выполняет несколько операций |
| 0 | Модуль, который высчитывает |
| 0 | Модуль, который записывает |
| 0 | несуществующий Модуль |
| 0 | Модуль, который не существует |

Вопрос №157

|  |  |
| --- | --- |
|  | **Какие есть 2 основных типа ПО?** |
| 1 | Системное и прикладное |
| 0 | Основное |
| 0 | Вспомогательное |
| 0 | Второстепенное |
| 0 | не существующее |

Вопрос №158

|  |  |
| --- | --- |
|  | **Что такое ПО?** |
| 1 | Неотъемлемая часть компьютерной системы |
| 0 | Часть диска |
| 0 | Диаграмма |
| 0 | Таблица |
| 0 | Картинка |

Вопрос №159

|  |  |
| --- | --- |
|  | **Чем управляет системное ПО?** |
| 1 | Компонентами компьютера |
| 0 | Таблицей |
| 0 | Диаграммой |
| 0 | Пикселем |
| 0 | Картинкой |

Вопрос №160

|  |  |
| --- | --- |
|  | **В виде чего могут быть представлены 2 вида ПО?** |
| 1 | В виде диаграммы |
| 0 | В виде 2-х точек |
| 0 | В виде таблицы |
| 0 | В виде картинки |
| 0 | В виде строк |

Вопрос №161

|  |  |
| --- | --- |
|  | **Сколько существует основных больших этапов разработки программного обеспечения?** |
| 1 | 5 |
| 0 | 2 |
| 0 | 3 |
| 0 | 4 |
| 0 | 6 |

Вопрос №162

|  |  |
| --- | --- |
|  | **Какой этап является первым при разработке программного обеспечения?** |
| 1 | постановка задачи |
| 0 | анализ требований и разработка спецификаций |
| 0 | проектирование |
| 0 | реализация |
| 0 | внедрение и эксплуатация |

Вопрос №163

|  |  |
| --- | --- |
|  | **Какой этап является вторым при разработке программного обеспечения?** |
| 1 | анализ требований и разработка спецификаций |
| 0 | постановка задачи |
| 0 | внедрение и эксплуатация |
| 0 | реализация |
| 0 | проектирование |

Вопрос №164

|  |  |
| --- | --- |
|  | **Какой этап является третьим при разработке программного обеспечения?** |
| 1 | проектирование |
| 0 | анализ требований и разработка спецификаций |
| 0 | постановка задачи |
| 0 | внедрение и эксплуатация |
| 0 | реализация |

Вопрос №165

|  |  |
| --- | --- |
|  | **Какой этап является четвертый при разработке программного обеспечения?** |
| 1 | реализация |
| 0 | проектирование |
| 0 | анализ требований и разработка спецификаций |
| 0 | постановка задачи |
| 0 | внедрение и эксплуатация |

Вопрос №166

|  |  |
| --- | --- |
|  | **Какой этап является заключающим при разработке программного обеспечения?** |
| 1 | внедрение и эксплуатация |
| 0 | реализация |
| 0 | проектирование |
| 0 | анализ требований и разработка спецификаций |
| 0 | постановка задачи |

Вопрос №167

|  |  |
| --- | --- |
|  | **Что определяют на этапе постановке задачи?** |
| 1 | Определяются назначение программного обеспечения и основные требования к нему. |
| 0 | Определяют конечный результат |
| 0 | Делают выводы о работе |
| 0 | Проходит работа непосредственно с готовым продуктом |
| 0 | Все ответы правильны |

Вопрос №168

|  |  |
| --- | --- |
|  | **Что определяют на этапе анализ требований и определение спецификаций?** |
| 1 | Требования и спецификации ПО |
| 0 | Обсуждения задач |
| 0 | Работа с ПО |
| 0 | Создание нового подхода |
| 0 | Все ответы правильны |

Вопрос №169

|  |  |
| --- | --- |
|  | **Что определяют на этапе проектирование?** |
| 1 | Все ответы правильны |
| 0 | разработка и уточнение структуры входных и выходных данных |
| 0 | создание алгоритмов решения задачи |
| 0 | описание форм представления входных и выходных данных |
| 0 | разработка структуры программы |

Вопрос №170

|  |  |
| --- | --- |
|  | **Что определяют на этапе реализация?** |
| 1 | Непосредственная разработка ПО |
| 0 | Постановка задачи |
| 0 | Получение конечного результата |
| 0 | Анализ требований |
| 0 | Все ответы правильны |

Вопрос №171

|  |  |
| --- | --- |
|  | **Что определяют на этапе внедрение и эксплуатация?** |
| 1 | Непосредственно конечный продукт и передача его заказчику |
| 0 | Отладка |
| 0 | Тестирование программы |
| 0 | Выбор материала для работы с ПО |
| 0 | Все ответы правильны |

Вопрос №172

|  |  |
| --- | --- |
|  | **Что нужно, для того, чтобы исключить проблемы, связанные с человеческим фактором?** |
| 1 | Все ответы правильны |
| 0 | Установление порядка выполнения действий |
| 0 | Определение состава и времени поставки артефактов, создаваемых в процессе разработки |
| 0 | Закрепление действий и артефактов за разработчиками |
| 0 | Введение критериев отслеживания хода проекта, измерение результатов и планирование будущих проектов |

Вопрос №173

|  |  |
| --- | --- |
|  | **С какими проблемами может столкнуться заказчик перед исполнителем?** |
| 1 | Все ответы правильны |
| 0 | Потребности заказчиков не поняты или не полностью зафиксированы |
| 0 | Заказчики не готовы выделить достаточно ресурсов под проект |
| 0 | Заказчики не стремятся к сотрудничеству с разработчиками |
| 0 | Ожидания заказчиков нереалистичны |

Вопрос №174

|  |  |
| --- | --- |
|  | **Сколько этапов существует в каскадной модели?** |
| 1 | 5 |
| 0 | 3 |
| 0 | 4 |
| 0 | 6 |
| 0 | 7 |

Вопрос №175

|  |  |
| --- | --- |
|  | **Как изменяется время при использование итерационной модели?** |
| 1 | Время сокращается |
| 0 | Время увеличивается |
| 0 | Время не изменяться |
| 0 | Все ответы правильны |
| 0 | Все ответы не правильны |

Вопрос №176

|  |  |
| --- | --- |
|  | **Сколько этапов в итерационной модели?** |
| 1 | 6 |
| 0 | 2 |
| 0 | 3 |
| 0 | 4 |
| 0 | 5 |

Вопрос №177

|  |  |
| --- | --- |
|  | **На какие этапы делиться спиральная модель?** |
| 1 | Все ответы правильны |
| 0 | Этап - Определение целей |
| 0 | Этап - Оценка и разрешение рисков |
| 0 | Этап - Разработка и тестирование |
| 0 | Этап - Планирование |

Вопрос №178

|  |  |
| --- | --- |
|  | **Модель быстрой разработки приложений появилась в …** |
| 1 | 1980-х гг. |
| 0 | 1990-х гг. |
| 0 | 1970-х гг. |
| 0 | 2000-х гг. |
| 0 | 2010-х гг. |

Вопрос №179

|  |  |
| --- | --- |
|  | **Сколько существует принципов в методе динамической разработки?** |
| 1 | 8 |
| 0 | 12 |
| 0 | 6 |
| 0 | 4 |
| 0 | 10 |

Вопрос №180

|  |  |
| --- | --- |
|  | **Какую классификацию моделей предлагает метод динамической разработки?** |
| 1 | Все ответы правильны |
| 0 | Коммерческие прототипы |
| 0 | Прототипы возможностей/прототипы проекта |
| 0 | Прототипы производительности |
| 0 | Прототипы применимости |

Вопрос №181

|  |  |
| --- | --- |
|  | **Укажите, какие классы используют представления сущностей реального мира или внутренних элементов системы, например, структур данных:** |
| 1 | Классы-сущности |
| 0 | Граничные классы |
| 0 | Управляющие классы |
| 0 | Управляемые классы |
| 0 | Исключения |

Вопрос №182

|  |  |
| --- | --- |
|  | **Укажите, какие классы служат для моделирования последовательного поведения, заложенного в один или несколько вариантов использования:** |
| 1 | Управляющие классы |
| 0 | Классы-сущности |
| 0 | Граничные классы |
| 0 | Исключения |
| 0 | Управляемые классы |

Вопрос №183

|  |  |
| --- | --- |
|  | **Укажите, какие классы обеспечивают взаимодействие между действующими лицами и внутренними элементами системы:** |
| 1 | Граничные классы |
| 0 | Управляющие классы |
| 0 | Исключения |
| 0 | Управляющие классы |
| 0 | Управляемые классы |

Вопрос №184

|  |  |
| --- | --- |
|  | **Какая диаграмма показывает, из каких частей состоит проектируемая программная система, и как эти части связаны друг с другом?** |
| 1 | Диаграмма пакетов |
| 0 | Диаграмма вариантов использования |
| 0 | Диаграмма классов |
| 0 | Диаграмма состояний объектов |
| 0 | Диаграмма кооперации |

Вопрос №185

|  |  |
| --- | --- |
|  | **Задачей какого проектирования при объектном подходе является разработка классов для реализации объектов, полученных при объектной декомпозиции, что предполагает полное описание полей и методов каждого класса?** |
| 1 | логического |
| 0 | компонентного |
| 0 | комплексного |
| 0 | физического |
| 0 | нет верного ответа |

Вопрос №186

|  |  |
| --- | --- |
|  | **Укажите, что лежит в основе объектного подхода к разработке ПО:** |
| 1 | объектная декомпозиция |
| 0 | декомпозиция сложных систем |
| 0 | объектный анализ |
| 0 | параллельное проектирование |
| 0 | создание конструкторов |

Вопрос №187

|  |  |
| --- | --- |
|  | **Какое проектирование при объектном подходе включает проектирование объединения классов и других программных ресурсов в программные компоненты и размещения этих компонентов на конкретных вычислительных установках.** |
| 1 | физическое |
| 0 | логическое |
| 0 | комплексное |
| 0 | компонентное |
| 0 | нет верного ответа |

Вопрос №188

|  |  |
| --- | --- |
|  | **Для выявления каких классов, как правило, изучают описания вариантов использования, концептуальную модель и диаграммы деятельностей?** |
| 1 | классов-сущностей |
| 0 | граничных классов |
| 0 | управляющих классов |
| 0 | исключений |
| 0 | управляемых классов |

Вопрос №189

|  |  |
| --- | --- |
|  | **Выделение памяти при создании объекта осуществляется при помощи специального метода класса -** |
| 1 | конструктора |
| 0 | деструктора |
| 0 | инициализации |
| 0 | завершения |
| 0 | описания |

Вопрос №190

|  |  |
| --- | --- |
|  | **Как называют пакеты, с которыми связаны все пакеты программной системы?** |
| 1 | глобальные |
| 0 | локальные |
| 0 | полные |
| 0 | связующие |
| 0 | диспетчеры |

Вопрос №191

|  |  |
| --- | --- |
|  | **Как называют классы, реализующие сценарии вариантов использования?** |
| 1 | Объекты управления |
| 0 | Объекты задачи |
| 0 | Интерфейс базы данных |
| 0 | Обработка ошибок |
| 0 | Базовые структуры данных |

Вопрос №192

|  |  |
| --- | --- |
|  | **Как называются классы, реализующие объекты предметной области системы?** |
| 1 | Объекты задачи |
| 0 | Объекты управления |
| 0 | Интерфейс базы данных |
| 0 | Обработка ошибок |
| 0 | Базовые структуры данных |

Вопрос №193

|  |  |
| --- | --- |
|  | **Классы, реализующие интерфейс с базой данных -** |
| 1 | Интерфейс базы данных |
| 0 | Объекты задачи |
| 0 | Объекты управления |
| 0 | Обработка ошибок |
| 0 | Базовые структуры данных |

Вопрос №194

|  |  |
| --- | --- |
|  | **Укажите классы, реализующие внутренние структуры данных, такие как деревья, п-связные списки и т. и.** |
| 1 | Базовые структуры данных |
| 0 | Интерфейс базы данных |
| 0 | Объекты задачи |
| 0 | Объекты управления |
| 0 | Обработка ошибок |

Вопрос №195

|  |  |
| --- | --- |
|  | **Для выявления каких классов изучают описания вариантов использования, концептуальную модель и диаграммы деятельностей?** |
| 1 | классов-сущностей |
| 0 | граничных классов |
| 0 | управляющих классов |
| 0 | исключений |
| 0 | управляемых классов |

Вопрос №196

|  |  |
| --- | --- |
|  | **Классы исключений, реализующие обработку нештатных ситуаций – это:** |
| 1 | Обработка ошибок |
| 0 | Базовые структуры данных |
| 0 | Интерфейс базы данных |
| 0 | Объекты задачи |
| 0 | Объекты управления |

Вопрос №197

|  |  |
| --- | --- |
|  | **Для обнаружения каких классов изучают пары «действующее лицо - вариант использования»?** |
| 1 | граничных классов |
| 0 | классов-сущностей |
| 0 | управляющих классов |
| 0 | исключений |
| 0 | управляемых классов |

Вопрос №198

|  |  |
| --- | --- |
|  | **При каком условии связь между пакетами фиксируют?** |
| 1 | если изменения в одном пакете могут повлечь за собой изменения в другом |
| 0 | если изменения в одном пакете не создают изменений в другом |
| 0 | в любом случае |
| 0 | связи между пакетами не фиксируются |
| 0 | связь между пакетами фиксируется случайным образом |

Вопрос №199

|  |  |
| --- | --- |
|  | **Какие диаграммы представляют собой схему потоков управления для решения некоторой задачи по отдельным дей­ствиям, допускают наличие параллельных и/или альтернативных действий?** |
| 1 | диаграммы деятельностей |
| 0 | диаграммы последовательностей действий |
| 0 | диаграммы кооперации |
| 0 | диаграммы размещения |
| 0 | диаграммы классов |

Вопрос №200

|  |  |
| --- | --- |
|  | **Как называются диаграммы, отображающие упорядоченное по времени взаимодействие объектов в процессе выполнения варианта использования?** |
| 1 | диаграммы последовательностей |
| 0 | диаграммы деятельностей |
| 0 | диаграммы последовательностей действий |
| 0 | диаграммы кооперации |
| 0 | диаграммы размещения |

Вопрос №201

|  |  |
| --- | --- |
|  | **Что является основной задачей логического проектирования при объектном подходе?** |
| 1 | Разработка классов для реализации объектов |
| 0 | Создание шаблона |
| 0 | Исследование структуры |
| 0 | Юнит тесты |
| 0 | Реализация архитектуры |

Вопрос №202

|  |  |
| --- | --- |
|  | **Большинство классов можно отнести к определенному типу, который применительно к классам называют**? |
| 1 | Стереотипом |
| 0 | Наследником |
| 0 | Структурой |
| 0 | Абстрактом |
| 0 | Все перечисленные |

Вопрос №203

|  |  |
| --- | --- |
|  | **Какие сообщения, не блокируют работу вызывающего объекта?** |
| 1 | Асинхронные |
| 0 | Синхронные |
| 0 | Последовательные |
| 0 | Постоянные |
| 0 | Все перечисленные |

Вопрос №204

|  |  |
| --- | --- |
|  | **Каким большим знаком показывают уничтожение объекта?** |
| 1 | Х |
| 0 | А |
| 0 | = |
| 0 | - |
| 0 | ! |

Вопрос №205

|  |  |
| --- | --- |
|  | **Укажите, альтернативный способ представления взаимодействия объектов в процессе реализации сценария** |
| 1 | Диаграммы кооперации |
| 0 | Диаграммы Венна |
| 0 | Диаграммы Ганта |
| 0 | Нет правильного ответа |
| 0 | Все перечисленные |

Вопрос №206

|  |  |
| --- | --- |
|  | **Укажите, как называются ассоциация между целым и его частью или частями?** |
| 1 | Агрегацией |
| 0 | Композиция |
| 0 | Декомпозиция |
| 0 | UML |
| 0 | Нет правильного ответа |

Вопрос №207

|  |  |
| --- | --- |
|  | **Укажите, более сильную разновидность агрегации, которая подразумевает, что объект-часть может принадлежать только единственному целому** |
| 1 | Композиция |
| 0 | Декомпозиция |
| 0 | Вертикальных |
| 0 | Агрегация |
| 0 | Все варианты верны |

Вопрос №208

|  |  |
| --- | --- |
|  | **Если стрелки указаны с обеих сторон, то это означает какую ассоциацию?** |
| 1 | Двунаправленную |
| 0 | Однонаправленную |
| 0 | Бинарную |
| 0 | Нет правильного ответа |
| 0 | Все варианты верны |

Вопрос №209

|  |  |
| --- | --- |
|  | **Получение из класса-шаблона класса с конкретными типами элементов называют?** |
| 1 | Связыванием |
| 0 | Наследованием |
| 0 | Инкапсуляцией |
| 0 | Полиморфизмом |
| 0 | Нет правильного ответа |

Вопрос №210

|  |  |
| --- | --- |
|  | Как в UML называют класс, содержащий только объявление операций? |
| 1 | Интерфейс |
| 0 | Класс А |
| 0 | Класс В |
| 0 | Буч |
| 0 | Коад |

Вопрос №211

|  |  |
| --- | --- |
|  | **Реализацию интерфейса также можно показать сколькими способами?** |
| 1 | Двумя |
| 0 | Тремя |
| 0 | Четырьмя |
| 0 | Все перечисленные |
| 0 | Нет правильного ответа |

Вопрос №212

|  |  |
| --- | --- |
|  | **Как называются основные действия, реализуемые классом?** |
| 1 | Операции |
| 0 | Выражения |
| 0 | Присваивание |
| 0 | Определения |
| 0 | Нет правильного ответа |

Вопрос №213

|  |  |
| --- | --- |
|  | **Сколько значений может принимать признак видимости?** |
| 1 | 3 |
| 0 | 2 |
| 0 | 1 |
| 0 | 4 |
| 0 | Нет правильного ответа |

Вопрос №214

|  |  |
| --- | --- |
|  | **Диаграмма компонентов оперирует какими понятиями?** |
| 1 | Компонент и зависимость |
| 0 | Компонент и независимость |
| 0 | Часть и зависимость |
| 0 | Часть и независимость |
| 0 | Нет правильного ответа |

Вопрос №215

|  |  |
| --- | --- |
|  | **Как называется отдельный экземпляр класса, который создается на этапе выполнения программы?** |
| 1 | Объект |
| 0 | Интерфейс |
| 0 | Составной класс |
| 0 | Структура |
| 0 | Нет правильного ответа |

Вопрос №216

|  |  |
| --- | --- |
|  | **Что означает значение “+” признака видимости?** |
| 1 | Общий |
| 0 | Частный |
| 0 | Независимый |
| 0 | Целый |
| 0 | Нет правильного ответа |

Вопрос №217

|  |  |
| --- | --- |
|  | **Как называют краткое неформальное перечисление основных функций объектов класса?** |
| 1 | Ответственностью класса |
| 0 | Состоянием класса |
| 0 | Диаграммой класса |
| 0 | Сборкой класса |
| 0 | Нет правильного ответа |

Вопрос №218

|  |  |
| --- | --- |
|  | **Диаграмму компонентов применяют при проектировании какой структуры разрабатываемого ПО?** |
| 1 | Физической |
| 0 | Абстрактной |
| 0 | Зависимой |
| 0 | Свободной |
| 0 | Нет правильного ответа |

Вопрос №219

|  |  |
| --- | --- |
|  | **Как называется спецификация существенного факта, который происходит во времени и пространстве?** |
| 1 | Событие |
| 0 | Задача |
| 0 | Функция |
| 0 | Интервал |
| 0 | Нет правильного ответа |

Вопрос №220

|  |  |
| --- | --- |
|  | **Как называется отношение между более общим элементом (родителем или предком) и более частным или специальным элементом (дочерним или потомком)?** |
| 1 | Обобщение |
| 0 | Композиция |
| 0 | Агрегация |
| 0 | Ассоциация |
| 0 | Зависимость |