|  |  |  |  |
| --- | --- | --- | --- |
| **Текст 16 А** *Псевдокод и объектно-ориентированное проектирование* Объектно-ориентированный дизайн связан с определением проблемы как набора объектов, а не набора функций. Объект можно рассматривать как контейнер для набора данных и операций, которые необходимо с ним выполнить. Объект создается на основе шаблона или шаблона, называемого классом, который определяет основные атрибуты и методы, доступные объектам этого класса. При создании объекта ему присваивается уникальное имя и копия всех атрибутов и методов из его класса. Процесс создания объектов из классов является называется инстанцированием. Из класса может быть создано множество объектов, каждый из которых содержит одни и те же атрибуты, но не обязательно хранит одни и те же значения в этих атрибутах. Процесс создания экземпляра объекта из шаблона класса выполняется специальным методом или набором инструкций, известным как конструктор. Пока конструктор не вызван, объект не существует. При создании каждого объекта он получает свою собственную копию всех атрибутов и методов для своего класса. Каждый класс должен иметь конструктор по умолчанию, который помещает фактические значения в атрибуты. Когда из класса создается несколько экземпляров объекта, важно, чтобы программа могла точно определить, какую копию метода она должна использовать при вызове метода. Это управляется с помощью специальной нотации, в которой имя объекта помещается перед методом, разделенным оператором. | **Текст 15 А** *Вирус против червя* Основное различие между вирусом и червем заключается в том, что вирусы должны запускаться активацией их хоста; в то время как черви-это автономные вредоносные программы, которые могут самостоятельно реплицироваться и распространяться независимо, как только они взломали систему. Черви не требуют активации или какого -либо вмешательства человека-для выполнения или распространения своего кода. Вирусы часто прикрепляются или скрываются в общих или загруженных файлах, как исполняемых файлах-программе, которая запускает скрипт, - так и неисполняемых файлах, таких как документ Word или файл изображения. Когда файл хоста принимается или загружается целевой системой, вирус остается бездействующим до тех пор, пока зараженный файл хоста не будет активирован. Только после активации файла хоста вирус может запускаться, выполняя вредоносный код и реплицируясь для заражения других файлов в вашей системе. В отличие от этого, черви не требуют активации своего хост-файла. Как только червь вошел в вашу систему, обычно через сетевое соединение или в виде загруженного файла, он может затем запускаться, самовоспроизводиться и распространяться без инициирующего события. A червь делает несколько копий самого себя, которые затем распространяются по сети или через подключение к Интернету. Эти копии будут заражать любые недостаточно защищенные компьютеры и серверы, которые подключаются-через сеть или Интернет - к первоначально зараженному устройству. | **Текст 14 А** Методы искусственного интеллекта (ИИ), разработанные для объединения данных, используют экспертные системы и нейронные сети. Экспертные системы-это компьютерные системы, предназначенные для эмуляции способности человеческого мозга принимать решения. Экспертные системы используют специализированные знания и опыт, и их часто называют системами, основанными на знаниях. Решение, принимаемое экспертной системой, будет основываться на информации, полученной в ходе ее разработки. Однако эффективность экспертной системы будет зависеть от суммы знаний, заранее запрограммированных в нем. Учитывая факты и данные, экспертные системы делают выводы о событии или гипотезе. Постоянство, последовательность, повышенная надежность, быстрая реакция и снижение затрат являются прямыми преимуществами использования экспертных систем. Искусственные нейронные сети (ННС) - это программные имитационные процессорные блоки, или узлы, которые обучаются для решения задач. Обучение ННС проводится с использованием исторических данных и связанных с ними результатов. NN вычисляет свой ответ на входные данные теста и сравнивает его с известным результатом. Нейронные сети могут быть очень полезны для решения задач в приложениях, где трудно указать алгоритм. Они обычно состоят из нескольких взаимосвязанных узлов или "нейронов", которые действуют как независимые единицы обработки. | **Текст 13 А**  Что такое объект в программировании? объектно-ориентированное программирование, или ООП, - это подход к решению задач , при котором все вычисления выполняются с использованием объектов. Объект-это компонент программы, который знает, как выполнять определенные действия и как взаимодействовать с другими элементами программы. Объекты являются основными единицами объектно - ориентированного программирования. Код в объектно-ориентированном программировании организован вокруг объектов. Как только у вас есть ваши объекты, они могут взаимодействовать друг с другом, чтобы что-то произошло. Допустим, вы хотите иметь программу, в которой человек садится в машину и едет на ней из пункта А в пункт Б. Вы бы начали с описания объектов, таких как человек и автомобиль. Это включает в себя методы: человек знает , как водить машину, а машина знает, каково это-быть за рулем. Как только у вас есть свои объекты, вы собираете их вместе, чтобы человек мог сесть в машину и уехать. Класс - это чертеж объекта. Вы можете думать о классе как о концепции, а объект является воплощением этой концепции. Вам нужен класс, прежде чем вы сможете создать объект. Итак, допустим, вы хотите использовать человека в своей программа. Вы хотите иметь возможность описать человека и заставить его что-то сделать. Класс под названием "человек" даст представление о том , как выглядит человек и что он может делать. Чтобы действительно использовать человека в своей программе, вам нужно создать объект. Класс person используется для создания объекта типа person. - Теперь вы можете описать этого человека и заставить его что-нибудь сделать. |
| **Текст 10 А** *Что такое компьютерный вирус?* Компьютерный вирус-это вредоносный фрагмент компьютерного кода, предназначенный для распространения с устройства на устройство. Подмножество вредоносных программ, эти угрозы самокопирования обычно предназначены для повреждения устройства или кражи данных. Подумайте о биологическом вирусе - таком, от которого вас тошнит. Он постоянно неприятен, мешает вам нормально функционировать и часто требует чего-то мощного, чтобы избавиться от него. Компьютерный вирус очень похож. Разработанные для безжалостной репликации, компьютерные вирусы заражают ваши программы и файлы, изменяют способ ваш компьютер работает или вообще перестает работать. Некоторые компьютерные вирусы запрограммированы на нанесение вреда вашему компьютеру, повреждая программы, удаляя файлы или переформатируя жесткий диск. Другие просто копируют себя или наводняют сеть трафиком, делая невозможным выполнение какой-либо интернет-активности. Еще менее вредные компьютерные вирусы могут значительно нарушить производительность вашей системы, истощая память компьютера и вызывая частые сбои компьютера. Даже если вы будете осторожны, вы можете подхватить компьютерные вирусы с помощью обычных веб -действий, таких как: • Обмен музыкой, файлами или фотографиями с другими пользователями \* Посещение зараженного веб - сайта • Открытие спама или вложения электронной почты • Загрузка бесплатных игр, панелей инструментов, медиаплееров и других системных утилит Установка основных программных приложений без тщательного чтения статей лицензионных соглашений/компьютер-безопасность-угрозы-компьютер | **Текст 11 А** Каков жизненный цикл разработки программного обеспечения? SDLC или Жизненный цикл разработки программного обеспечения-это процесс, который производит программное обеспечение с высочайшим качеством и наименьшими затратами в кратчайшие сроки. SDLC обеспечивает хорошо структурированный поток этапов, которые помогают организации быстро производить высококачественное программное обеспечение, которое хорошо протестировано и готово к использованию в производстве. SDLC включает в себя шесть этапов, как описано во введении. Популярные модели SDLC включают модель водопада, спиральную модель и гибкую модель. SDLC работает за счет снижения затрат на разработку программного обеспечения, в то время как одновременно улучшая качество и сокращая время производства. SDLC достигает этих явно расходящихся целей, следуя плану, который устраняет типичные подводные камни проектов разработки программного обеспечения. Этот план начинается с оценки существующих систем на предмет наличия недостатков. Далее в нем определяются требования к новой системе. Затем он создает программное обеспечение на этапах анализа, планирования, проектирования, разработки, тестирования и развертывания. Предвидя дорогостоящие ошибки, такие как неспособность запросить обратную связь у конечного пользователя или клиента, SLDC может устранить избыточную переработку и исправления после факта. Также важно знать, что на этапе тестирования уделяется большое внимание. Поскольку SDLC-это повторяющаяся методология, вы должны обеспечивать качество кода на каждом цикле. Многие организации, как правило, тратят мало усилий на тестирование, в то время как более пристальное внимание к тестированию может сэкономить им много переделок, времени и денег. Будьте умны и пишите правильные типы тестов. | *Как Происходит Заражение Вредоносными Программами?* **Текст 12 А** Авторы вредоносных программ используют различные физические и виртуальные средства для распространения вредоносных программ, которые заражают устройства и сети. Например, вредоносные программы могут быть доставлены в систему с помощью USB-накопителя или могут распространяться через Интернет посредством загрузки с диска, которая автоматически загружает системы вредоносных программ без ведома пользователя. Фишинговые атаки-это еще один распространенный тип доставки вредоносных программ, когда электронные письма, замаскированные под законные сообщения, содержат вредоносные ссылки или вложения, которые могут доставить исполняемый файл вредоносного ПО ничего не подозревающим пользователям. Сложные атаки вредоносных программ часто включают использование сервера управления и управления , который позволяет субъектам угроз взаимодействовать с зараженными системами, эксфильтратировать конфиденциальные данные и даже удаленно управлять скомпрометированным устройством или сервером. Новые штаммы вредоносных программ включают в себя новые методы уклонения и запутывания, которые предназначены не только для того, чтобы обмануть пользователей, но и администраторов безопасности , а также продукты защиты от вредоносных программ. Некоторые из этих методов уклонения основаны на простых тактиках, таких как использование веб-прокси для скрытия вредоносного трафика или исходного IP-адреса адреса. Более изощренные угрозы включают полиморфные вредоносные программы, которые могут неоднократно менять свой базовый код, чтобы избежать обнаружения от сигнатурный средства обнаружения; анти-песочница методов, которые позволяют обеспечением для обнаружения когда выполняется анализ, и, чтобы отложить выполнение до тех пор, пока листья песочница; и fileless Malware, которая проживает только в ОЗУ системы, чтобы избежать будучи обнаружен. | **ТЕКСТ 9А** 4 Современных Языка Программирования, которые работают медленно Существует более 700 различных языков программирования, и из этих языков только 250 удалось остаться на плаву. Остальные были выброшены сообществом программистов. Каждый язык программирования является формальным языком, используемым в компьютерном программировании для реализации алгоритмов. В то время как производительность языка всегда была главной предпосылкой для программистов, они всегда стараются прийти с новыми достижениями. Если программист создает языковое программное обеспечение, мы должны знать, что каждый язык реализован по-разному. Поэтому программе, которая должна будет выполнять тяжелые задачи, потребуется более быстрый язык программирования. Есть четыре самых популярных языка программирования, которые работают медленно. 1. Python • Python-самый популярный язык программирования, используемый в науке, веб -разработке и кибербезопасности. • Но Python, к сожалению, работает медленно из-за GIL' и факта использования интерпретатора 2. PHP2 • PHP-это язык веб-разработки, который используется для создания динамического веб-сайта. • Хотя язык PHP используется 79,0% всех веб-сайтов, но это невероятно медленно из-за отсутствия своевременной компиляции 3. Рубин • Ruby в основном используется для создания веб-приложений. • Ruby также страдает от высокого потребления памяти и медленного сбора мусора 4. Perl • Perl-это язык программирования общего назначения, первоначально разработанный для обработки тестов. • Он используется для широкого спектра задач, таких как разработка системы , сетевое программирование, графический интерфейс\* разработка и многое другое Я довольно чувствителен к новым реализациям |
| **Текст 8 А**  *Тенденция развития процессорных технологий* Развитие процессорной технологии прошло более чем 40 лет. Ее дорожная карта развития сместился масштабирования для масштабирования и целью резко отодвигается от преследования с более высокой тактовой частотой, а вместо этого направлен на создание более ядер на каждом процессоре, согласно закону Мура, нажимая вычислительной частота процессора постоянно является одним из важнейших способов повысить производительность компьютера в эпоху последовательных вычислений, в то же время, лоты методов оптимизации, таких как параллелизм на уровне команд (ILP), конвейер, предварительная выборка, предсказание ветвления, выполнение команд вне порядка, многоуровневый кэш и гиперпоточность могут быть автоматически идентифицированы и использованы процессором и компилятором. Высокопроизводительные процессорные технологии вступили в эру многоядерных процессоров, и многоядерные технологии параллельной обработки стали основным направлением. Но хотя возможности обработки данных были значительно расширены в многоядерных архитектурах, программное обеспечение не может автоматически получать преимущества. Вместо, программисты должны преобразовать традиционные последовательные программы в параллельные программы и оптимизировать производительность алгоритма для LLC (последний уровень Кэш) многоядерных процессоров. В настоящее время производительность многоядерных процессоров значительно улучшилась с помощью полупроводниковых технологий. | **Текст 17А**  *Что такое контроль версий?* Как контроль версий помогает процветать высокопроизводительным командам разработчиков и разработчиков. Управление версиями, также известное как управление версиями, - это практика отслеживания и управления изменениями в программном коде. Системы контроля версий - это программные средства, которые помогают командам разработчиков программного обеспечения управлять изменениями исходного кода с течением времени. Поскольку среды разработки ускорились, системы контроля версий помогают командам разработчиков работать быстрее и умнее. Они особенно полезны для команд DevOps, поскольку помогают им сократить время разработки и увеличить количество успешных развертываний. Программное обеспечение для контроля версий отслеживает каждую модификацию кода в специальной базе данных. Если допущена ошибка, разработчики могут повернуть время вспять и сравнить более ранние версии кода, чтобы помочь исправить ошибку, сведя к минимуму сбои для всех членов команды. Почти для всех программных проектов исходный код подобен драгоценностям короны - драгоценному активу, ценность которого должна быть защищена. Для большинства команд разработчиков исходный код является хранилищем бесценных знаний и понимания проблемной области, которые разработчики собрали и усовершенствовали благодаря тщательным усилиям. Контроль версий защищает исходный код как от катастрофы, так и от случайной деградации человеческой ошибки и непреднамеренных последствий. | **Текст 1А**  *Характеристики больших данных* Большие данные можно описать следующими характеристиками: () Объем - Само название Больших данных связано с огромным размером . Могут ли конкретные данные на самом деле рассматриваться как большой Данные или нет, зависит от объема данных.  (ii) Разнообразие - Следующий аспект Больших данных-это их разнообразие. Разнообразие относится к разнородным источникам и характеру данных, как структурированных, так и неструктурированных. В настоящее время данные в виде электронных писем, видео и т.д. Также рассматриваются в приложениях для анализа. Это разнообразие неструктурированные данные создают определенные проблемы для хранения, интеллектуального анализа и анализа данных. Как быстро генерируются и обрабатываются данные для удовлетворения потребностей, (iii) Скорость - Термин "скорость" относится к скорости генерации данных. определяет реальный потенциал в данных. (iii) Изменчивость - Это относится к несоответствию, которое иногда может проявляться в данных, что затрудняет процесс эффективной обработки данных и управления ими. Способность обрабатывать большие данные приносит множество преимуществ, таких как: • предприятия могут использовать внешнюю разведку при принятии решений; улучшенное обслуживание клиентов; • раннее выявление рисков, связанных с продуктом/услугами, если таковые имеются.; • повышение операционной эффективности. Технологии больших данных можно использовать для создания промежуточной области или зоны посадки для новых данных, прежде чем определить, какие данные следует переместить в хранилище данных |  |