**Билет 1 МДК04.01**

**Теоретический вопрос:**

**Методы восстановления системы в Windows 10. Какие методы восстановления системы предусмотрены в Windows 10, как они работают, и в каких случаях их использование наиболее эффективно? Подкрепите свой ответ примерами, демонстрирующими устранение сбоев с помощью точек восстановления и образов системы.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:** В Windows 10 доступны несколько методов восстановления системы, каждый из которых применяется в зависимости от характера сбоя:

1. **Точки восстановления системы:** Точки восстановления позволяют откатить системные файлы и настройки к стабильному состоянию. Это полезно при ошибках, вызванных обновлениями, драйверами или изменением системных настроек.

**Пример:** После установки драйвера для новой видеокарты система начала выдавать синий экран смерти (BSOD). Использование точки восстановления, созданной до установки драйвера, вернуло систему в рабочее состояние.

**Как использовать:** Перейдите в "Панель управления" → "Система" → "Защита системы". Запустите "Восстановление системы" и выберите точку восстановления.

1. **Образ системы:** Образ системы — это полная копия операционной системы, её данных и настроек, создаваемая для восстановления после серьёзных сбоев. **Пример:** После повреждения файловой системы Windows 10 перестала загружаться. Заранее созданный образ позволил восстановить систему до работоспособного состояния. **Как использовать:** Создайте образ через "Панель управления" → "Резервное копирование и восстановление". Для восстановления загрузитесь с установочного носителя Windows и выберите "Восстановление образа системы".

Эти методы помогают минимизировать последствия сбоев и поддерживать стабильность системы.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите для этого пользователя изменение фона рабочего стола.

**Билет 2 МДК04.01**

**Теоретический вопрос:**

**Методы восстановления системы в Linux Mint. Какие методы восстановления системы предусмотрены в Linux Mint, как они работают, и как их применение помогает устранить сбои? Подкрепите свой ответ примерами, демонстрирующими использование Timeshift и восстановление загрузчика GRUB.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:** В Linux Mint существуют инструменты для восстановления системы, позволяющие исправлять различные виды сбоев:

1. **Timeshift:** Это встроенный инструмент для создания "снимков" системы, которые можно использовать для отката к предыдущему состоянию. **Пример:** После установки обновлений Linux Mint начала работать нестабильно. Откат системы к последнему стабильному снимку, созданному Timeshift, решил проблему. **Как использовать:** Установите Timeshift командой sudo apt install timeshift. Создайте снимок и используйте его для восстановления через интерфейс программы.
2. **Восстановление загрузчика GRUB:** GRUB отвечает за загрузку Linux Mint. Если он повреждён (например, после установки Windows), система не загружается. **Пример:** Установка Windows на компьютер с Linux перезаписала GRUB, что сделало Linux недоступным. Восстановление загрузчика через Live USB вернуло GRUB и доступ к обеим ОС. **Как восстановить:** Загрузитесь с загрузочной флешки Linux Mint. Смонтируйте корневой раздел: sudo mount /dev/sdXn /mnt. Восстановите GRUB: sudo grub-install --root-directory=/mnt /dev/sdX.

Эти методы позволяют эффективно решать проблемы, связанные с программными и загрузочными сбоями, обеспечивая стабильность системы.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите для этого пользователя включение заставки.

**Билет 3 МДК04.01**

**Теоретический вопрос:**

**Безопасный режим как инструмент диагностики программного обеспечения. Что такое безопасный режим, как он используется для тестирования совместимости программного обеспечения, и какие преимущества он имеет по сравнению с обычным режимом работы системы? Приведите примеры устранения проблем с драйверами и конфликтующими программами.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:** Безопасный режим — это специальный режим работы операционной системы, при котором загружается минимальный набор драйверов и системных компонентов. Этот режим используется для диагностики и устранения проблем, связанных с конфликтами программного обеспечения, драйверов или системных компонентов.

**Роль в тестировании ПО:** Безопасный режим позволяет исключить влияние сторонних программ и драйверов, что упрощает поиск источника проблемы. В этом режиме можно проверить, как программа взаимодействует с базовыми компонентами ОС.

**Пример 1:** Устранение проблем с драйверами. Допустим, драйвер видеокарты вызывает сбои при запуске приложения для рендеринга 3D-графики. В безопасном режиме драйвер видеокарты не загружается, что позволяет запустить приложение без сбоев. После выявления проблемы администратор может удалить или обновить драйвер через диспетчер устройств.

**Пример 2:** Устранение конфликтов программ. Программа для защиты данных блокирует работу интернет-браузера. В безопасном режиме эта программа не запускается, что позволяет браузеру работать корректно. После этого администратор может отключить автозапуск программы или удалить её.

Преимущества безопасного режима:

1. Минимизация влияния сторонних программ и драйверов.
2. Возможность изоляции проблемных компонентов.
3. Удобство использования встроенных инструментов диагностики, таких как диспетчер устройств и журнал событий.

Безопасный режим — это эффективный инструмент для диагностики проблем совместимости, позволяющий локализовать источник проблем и устранить их с минимальными последствиями.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, скройте и отключите все элементы рабочего стола.

**Билет 4 МДК04.01**

**Теоретический вопрос:**

**Инструменты диагностики и устранения проблем, выявленных в безопасном режиме. Какие системные и сторонние инструменты используются для исправления проблем, обнаруженных в безопасном режиме? Приведите примеры использования диспетчера устройств, журнала событий Windows и программы Autoruns.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

После выявления проблемы в безопасном режиме следующим шагом является её устранение с использованием различных инструментов диагностики.

**1. Диспетчер устройств:** Этот инструмент позволяет обновлять, отключать или удалять драйверы, которые вызывают проблемы.

**Пример:** Если драйвер сетевой карты несовместим с новым приложением, администратор может удалить драйвер через диспетчер устройств и установить его обновлённую версию.

**2. Журнал событий Windows:** Этот инструмент помогает анализировать ошибки, которые произошли до перехода в безопасный режим. Он предоставляет информацию о сбоях драйверов или приложений.

**Пример:** В журнале событий администратор обнаруживает ошибку, связанную с драйвером принтера. Это указывает на необходимость обновления или удаления драйвера.

**3. Программа Autoruns (сторонний инструмент):** Этот инструмент позволяет отключать автоматическую загрузку проблемных программ, которые могут вызывать конфликты.

**Пример:** Программа антивируса конфликтует с другой утилитой, вызывая системные сбои. С помощью Autoruns администратор отключает автозапуск антивируса, что позволяет изолировать проблему.

**Дополнительные инструменты:**

* **Программа «Установка и удаление программ»**: используется для удаления конфликтующих программ.
* **Средства восстановления системы**: точки восстановления помогают откатить систему к стабильному состоянию.

Эти инструменты позволяют эффективно устранять проблемы совместимости, выявленные в безопасном режиме, и обеспечивают надёжность работы системы.

***https://learn.microsoft.com/ru-ru/sysinternals/downloads/autoruns***

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, скройте значок Компьютер с рабочего стола.

**Билет 5 МДК04.01**

**Теоретический вопрос:**

**Групповые политики в Windows. Что такое групповые политики, для чего они используются, и как они помогают централизованно управлять конфигурацией операционной системы? Приведите примеры настройки безопасности и автоматизации через групповые политики.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Групповые политики (Group Policy) — это механизм в операционных системах Windows, который позволяет администраторам централизованно управлять настройками конфигурации и безопасности системы, а также программного обеспечения. Политики могут применяться как к отдельным компьютерам, так и к группам пользователей, что значительно упрощает администрирование в корпоративной среде.

**Основные функции групповых политик:**

1. **Управление конфигурацией системы**: Позволяют настраивать параметры ОС, такие как доступ к панели управления, параметры сети и поведения системы.
2. **Обеспечение безопасности**: Политики безопасности включают шифрование данных, настройку брандмауэра, требование сложных паролей.
3. **Автоматизация управления**: С помощью групповых политик можно автоматизировать задачи, такие как установка программного обеспечения или обновление драйверов.

**Пример 1: Настройка безопасности через групповые политики**

Если необходимо повысить безопасность системы, можно настроить требование сложных паролей:

* Откройте Local Group Policy Editor (Win+R → gpedit.msc).
* Перейдите в **Конфигурация компьютера → Параметры Windows → Параметры безопасности → Политики учетных записей → Политика паролей**.
* Включите политику **Требовать сложные пароли**.

**Пример 2: Автоматическая блокировка экрана**

Чтобы уменьшить риск доступа к системе посторонних лиц:

* Перейдите в **Конфигурация пользователя → Административные шаблоны → Панель управления → Персонализация**.
* Включите параметр **Ограничение времени ожидания блокировки экрана** и задайте значение (например, 5 минут).

Групповые политики позволяют администраторам централизованно управлять настройками системы, минимизируя ошибки и повышая эффективность работы всей сети.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите свойства из контекстного меню значка Компьютер.

**Билет 6 МДК04.01**

**Теоретический вопрос:**

**Установочный образ операционной системы. Что такое установочный образ ОС, зачем его модифицировать, и какие инструменты позволяют изменять настройки по умолчанию? Продемонстрируйте на примере изменение языка и интеграцию драйверов в образ Windows 10 Pro.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Установочный образ операционной системы (ISO-файл) — это файл, содержащий все данные, необходимые для установки операционной системы. Он включает драйверы, системные библиотеки, настройки и файлы операционной системы. Модификация установочного образа позволяет настроить ОС перед её установкой, что удобно для массового развёртывания или специфических требований.

**Зачем модифицировать образ?**

1. **Изменение настроек языка и региона:** Например, если изначально язык системы установлен на английский, но нужно использовать русский язык, то это можно сделать заранее, изменив настройки в образе.
2. **Интеграция драйверов и обновлений:** Это необходимо, чтобы операционная система поддерживала новое оборудование или сразу содержала актуальные обновления.

**Инструменты для модификации:**

* **DISM (Deployment Image Servicing and Management Tool):** DISM позволяет монтировать образ ОС, изменять настройки и добавлять компоненты.
* **oscdimg:** Используется для создания нового ISO-образа с изменёнными параметрами.

**Пример 1: Изменение языка в образе Windows 10 Pro**

1. Смонтируйте файл install.wim с помощью команды:

dism /Mount-Wim /WimFile:C:\Win10ProSetup\sources\install.wim /index:1 /MountDir:C:\mount

1. Измените язык интерфейса на русский:

dism /Image:C:\mount /Set-AllIntl:ru-RU

1. Сохраните изменения:

dism /Unmount-Wim /MountDir:C:\mount /Commit

**Пример 2: Добавление драйверов**

Для интеграции драйверов выполните:

dism /Image:C:\mount /Add-Driver /Driver:C:\drivers\ /Recurse

Эти изменения позволят установить ОС с заранее настроенными параметрами, что упрощает развёртывание в корпоративной среде или для специфических задач.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите корзину с рабочего стола.

**Билет 7 МДК04.01**

**Теоретический вопрос:**

**Подключение к сетевым ресурсам. Что такое сетевые ресурсы, как к ним подключиться в Windows и Linux, и какие шаги необходимо выполнить для подключения? Продемонстрируйте на примере подключение к сетевым ресурсам с использованием консольных команд.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Сетевые ресурсы — это файлы, папки, принтеры или серверы, которые доступны пользователям через локальную или глобальную сеть. Доступ к сетевым ресурсам позволяет совместно использовать данные и устройства между несколькими компьютерами. В качестве примера можете привести сетевой диск S: на котором хранятся лекции по предметам, задания и отчеты по лаб.работам, программное обеспечение. Данный ресурс централизованно используется как студентами, так и преподавателями для совместной работы и обмена информацией (файлами).

**Подключение к сетевым ресурсам в Windows**

1. **Через командную строку:** Используйте команду net use:

net use Z: \\server\share /user:username password

Где Z: — буква диска, \\server\share — путь к ресурсу, username и password — учётные данные.

**Подключение к сетевым ресурсам в Linux**

1. **Через терминал:**
   * Для подключения к папке используйте команду:

sudo mount -t cifs //server/share /mnt/share -o username=user,password=pass

Где /mnt/share — это точка монтирования.

Эти команды позволяют эффективно подключаться к сетевым ресурсам, предоставляя доступ к файлам или папкам на удалённых серверах.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите блокировку компьютера при нажатии комбинации кнопок Ctrl+Alt+Del.

**Билет 8 МДК04.01**

**Теоретический вопрос:**

**Модули обеспечения совместимости программного обеспечения. Что такое модуль обеспечения совместимости, для чего он используется, и какие основные функции он выполняет? Приведите примеры использования модулей совместимости для старых игр и виртуализации устаревших платформ.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Модуль обеспечения совместимости — это программный компонент, предназначенный для устранения проблем взаимодействия между разными версиями операционных систем, библиотек, драйверов и аппаратных компонентов. Он позволяет старым программам работать на новых платформах или обеспечивает взаимодействие между программами, созданными для разных сред.

**Основные функции модулей совместимости:**

1. **Эмуляция устаревших функций или API:** Это позволяет программам использовать функции, которые были удалены или изменены в новой версии ОС. **Пример:** Wine для Linux эмулирует вызовы Windows API, чтобы запускать Windows-программы в Linux.
2. **Конвертация данных:** Модули совместимости преобразуют данные из одного формата в другой для обеспечения работы между разными программами.
3. **Интеграция с драйверами:** Модуль обеспечивает связь между устаревшими драйверами и новыми компонентами ОС.

**Пример 1: Модуль совместимости для старых игр.** Многие старые игры, разработанные для Windows XP, не запускаются на Windows 10. Для решения этой проблемы используется режим совместимости, который эмулирует старые версии библиотек DirectX и файловых систем.

**Пример 2: Виртуализация устаревших платформ.** Для работы программ, созданных для Windows 98, можно использовать виртуальные машины, такие как VirtualBox или VMware. Они создают изолированную среду, эмулирующую устаревшую платформу, что позволяет запускать старое ПО без изменения кода.

Также можете рассказать про DOSBox, который эмулирует архитектуру x86, включая старые процессоры, такие как Intel 286 или 386, что позволяет запускать программы, которые не поддерживаются современными процессорами. Поддерживаются устаревшие звуковые карты, такие как SoundBlaster, и графические режимы VGA. Это важно для старых игр и мультимедийных программ, которые полагались на конкретное аппаратное обеспечение.

**Сравнение с другими инструментами:**

DOSBox отличается от других модулей совместимости тем, что он не использует нативный код операционной системы. Вместо этого он полностью эмулирует устаревшее оборудование, что делает его идеальным для запуска программ, которые полагаются на специфические аппаратные возможности старых систем.

Ну и закончим тем (или подведем итог), что модули совместимости помогают увеличить срок службы программного обеспечения, снижая затраты на его обновление и переадаптацию под актуальные версии ОС и железа.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите диспетчер задач из списка действий при нажатии комбинации кнопок Ctrl+Alt+Del.

**Билет 9 МДК04.01**

**Теоретический вопрос:**

**Работа модулей обеспечения совместимости. Как работают модули совместимости, и какие шаги выполняются при преобразовании устаревших системных вызовов? Продемонстрируйте это на примере Wine и режима совместимости в Windows.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Модули совместимости функционируют как промежуточный слой между программным обеспечением и операционной системой. Они интерпретируют команды старых программ и преобразуют их в команды, поддерживаемые новой ОС. Этот процесс включает следующие шаги:

**Основные шаги работы модуля совместимости:**

1. **Перехват системных вызовов:** Модуль улавливает обращения программы к функциям ОС, которые больше не поддерживаются.
2. **Трансляция команд:** Перехваченные вызовы преобразуются в новые команды или функции, которые поддерживаются текущей ОС.
3. **Передача ОС:** Преобразованные команды отправляются в операционную систему для выполнения.

**Пример 1:** Wine (Windows эмулятор в Linux) преобразует вызовы Windows API в вызовы Linux.

**Процесс работы:**

* Когда программа обращается к Windows API, Wine перехватывает этот вызов.
* Преобразует его в эквивалентный вызов Linux (например, графические команды DirectX переводятся в OpenGL).
* Команда передаётся в Linux для выполнения. Это позволяет запускать Windows-программы, такие как игры или офисные приложения, на Linux.

**Пример 2: Режим совместимости Windows.** Этот встроенный инструмент позволяет запускать старые программы на новых версиях Windows.

**Процесс работы:**

* При активации режима совместимости ОС эмулирует окружение более старой версии Windows, включая устаревшие библиотеки.
* Например, программа, созданная для Windows XP, может использовать устаревшие вызовы API через эмуляцию.

Эти механизмы обеспечивают работу устаревшего ПО в новых условиях, минимизируя проблемы совместимости.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите выход из системы при нажатии комбинации кнопок Ctrl+Alt+Del.

**Билет 10 МДК04.01**

**Теоретический вопрос:**

**Совместимость программного обеспечения. Что такое совместимость программного обеспечения, какие виды совместимости существуют, и как они обеспечиваются? Объясните на примерах прямой, обратной и кроссплатформенной совместимости.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Совместимость программного обеспечения — это способность программы работать корректно в различных условиях, учитывая оборудование, операционную систему, библиотеки и другие программные компоненты. Совместимость играет ключевую роль для стабильной и эффективной работы ПО.

**Основные виды совместимости:**

1. **Прямая совместимость:** Программа работает без изменений в своём изначальном окружении. **Пример:** текстовый редактор, разработанный для Windows 10, работает на этой ОС без дополнительных настроек.
2. **Обратная совместимость:** Новая версия программы или операционной системы поддерживает работу с файлами или программами, созданными в старых версиях. **Пример:** Microsoft Office позволяет открывать и редактировать документы, созданные в старых версиях Word.
3. **Кроссплатформенная совместимость:** Программа может работать на разных операционных системах. **Пример:** браузер Google Chrome доступен на Windows, macOS, Linux, а также мобильных платформах, что делает его универсальным.

**Как обеспечивается совместимость?**

**Тестирование на различных конфигурациях:** Производители программного обеспечения проводят тестирование своих продуктов в различных окружениях, чтобы убедиться в совместимости с разными версиями ОС, аппаратным обеспечением и драйверами. Тестирование может включать использование виртуальных машин, эмуляторов и реального оборудования.

**Использование универсальных стандартов и технологий:** Применение стандартов, таких как HTML5, OpenGL или универсальных языков программирования (Python, Java), обеспечивает независимость от конкретного оборудования и операционных систем. Использование API и SDK от производителей также помогает в создании совместимого ПО.

**Регулярные обновления:** Обновления позволяют ПО адаптироваться к изменениям в операционной системе, драйверах и аппаратных компонентах. Патчи выпускаются для устранения проблем совместимости с новыми версиями Windows или macOS.

**Эмуляция и виртуализация:** Для обеспечения работы старого ПО на современных системах используются эмуляторы, такие как Wine, или виртуальные машины, которые создают изолированное окружение.

**Обратная связь от пользователей:** Производители собирают данные о проблемах, возникающих у пользователей, и выпускают обновления для устранения ошибок и несовместимостей.

Совместимость программного обеспечения позволяет избежать конфликтов и повысить удобство использования ПО в различных средах.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики,

**Билет 11 МДК04.01**

**Теоретический вопрос:**

**Совместимость драйверов. Почему драйверы играют ключевую роль в совместимости, и какие проблемы могут возникнуть при использовании несовместимых драйверов? Подкрепите ответ примерами обновления и исправления драйверов.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Драйверы — это программные компоненты, которые обеспечивают взаимодействие операционной системы с аппаратным обеспечением, такими как принтеры, видеокарты и сетевые адаптеры. Совместимость драйверов важна для корректной работы всех устройств, подключённых к компьютеру.

**Почему драйверы важны?**

1. Они выступают посредниками между операционной системой и аппаратными компонентами.
2. Обеспечивают стабильность и производительность системы.
3. Позволяют использовать современные функции оборудования.

**Проблемы с несовместимостью драйверов:**

1. **Устаревшие драйверы:** Если драйверы не обновлены для новой ОС, устройство может работать с ошибками или вовсе не работать. **Пример:** видеокарта NVIDIA без обновлённых драйверов может показывать графические артефакты в новых играх.
2. **Неправильный драйвер:** Установка несовместимого драйвера может привести к сбоям. **Пример:** использование неправильного драйвера сетевого адаптера может полностью отключить доступ в интернет.
3. **Конфликт между драйверами:** Несовместимые версии драйверов могут вызывать конфликты, влияющие на производительность системы. **Пример:** конфликт драйверов звуковой карты и видеокарты может вызывать сбои в воспроизведении звука.

**Как избежать проблем?**

1. Регулярно обновлять драйверы через встроенные инструменты (например, Windows Update) или сайты производителей. **Пример:** обновление драйвера NVIDIA через GeForce Experience для повышения производительности в новых играх.
2. Использовать сертифицированные драйверы, чтобы избежать ошибок.
3. Проводить тестирование после обновлений драйверов, чтобы убедиться в их совместимости с системой.

Совместимость драйверов напрямую влияет на стабильность и функциональность компьютера, поэтому их обновление и использование правильных версий критически важно.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, настройте запуск Far Manager при входе в систему

**Билет 12 МДК04.01**

**Теоретический вопрос:**

**Причины возникновения проблем совместимости программного обеспечения. Какие основные факторы влияют на совместимость ПО, как они проявляются, и каким образом обновления оборудования или библиотек могут стать причиной проблем? Продемонстрируйте это на примерах из различных ситуаций.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Проблемы совместимости программного обеспечения могут возникать из-за различных изменений в аппаратной и программной среде. Основные причины включают:

1. **Обновление аппаратного обеспечения:** Новое оборудование может использовать архитектуру или функции, которые не поддерживаются старым ПО. **Пример 1:** Переход на ARM-процессоры, такие как в ноутбуках с Windows на ARM, делает несовместимым ПО, написанное для архитектуры Intel. **Пример 2:** Программы, оптимизированные под HDD, могут работать некорректно при использовании SSD, например, восстановление данных.
2. **Обновление операционных систем:** Новые версии ОС часто убирают устаревшие функции или компоненты, что делает старые программы неработоспособными. **Пример 1:** Прекращение поддержки 32-битных приложений в macOS Catalina сделало многие программы несовместимыми. **Пример 2:** Программы, созданные для Windows 7, могут не работать на Windows 10 из-за изменений в API.
3. **Изменение библиотек и фреймворков:** Программы, использующие устаревшие версии библиотек, могут перестать работать с новыми обновлениями. **Пример:** Устаревшие функции в старой версии Java могут быть удалены в новой версии JRE.
4. **Аппаратные драйверы:** Несовместимые или устаревшие драйверы могут вызывать сбои или невозможность использования оборудования. **Пример:** Принтер перестаёт работать после обновления Windows, если драйверы не обновлены.
5. **Конфликты между программами:** Использование одинаковых библиотек или ресурсов может вызвать сбои. **Пример:** Два антивируса на одном устройстве конфликтуют, блокируя работу друг друга.

Эти причины требуют от разработчиков и пользователей тщательного подхода к тестированию и настройке ПО.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, настройте запуск Notepad++ при входе в систему

**Билет 13 МДК04.01**

**Теоретический вопрос:**

**Методы выявления проблем совместимости программного обеспечения. Какие подходы и инструменты применяются для обнаружения проблем совместимости ПО, и как они работают на практике в различных сценариях?**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Выявление проблем совместимости является важным этапом разработки и эксплуатации ПО. Основные методы включают:

1. **Тестирование на разных платформах:** Это основной способ проверки работы ПО в различных средах. **Пример 1:** Разработчики игр тестируют свои продукты на разных конфигурациях видеокарт, процессоров и ОС, чтобы обеспечить стабильность. **Пример 2:** При создании мобильных приложений их тестируют на Android и iOS, чтобы выявить различия в поведении.
2. **Использование виртуальных машин и контейнеров:** Эти инструменты позволяют симулировать различные аппаратные и программные среды. **Пример 1:** VirtualBox используется для тестирования приложений в Linux на машине с Windows. **Пример 2:** Docker помогает изолировать окружение и проверять взаимодействие разных библиотек.
3. **Инструменты анализа совместимости:** Автоматические утилиты помогают выявить проблемы до установки ПО. **Пример 1:** AppCompat Toolkit от Microsoft помогает проверить совместимость приложений с Windows. **Пример 2:** Windows Compatibility Center позволяет пользователям убедиться в том, что их устройства и приложения совместимы с системой.
4. **Анализ логов и отчетов о сбоях:** Журналы позволяют точно определить, какие компоненты вызывают проблемы. **Пример 1:** В Windows системный журнал помогает понять, какой драйвер вызвал "синий экран смерти". **Пример 2:** Программа Console в macOS предоставляет данные о сбоях запускаемых программ.
5. **Использование стандартов и API:** Программы, использующие общепринятые стандарты, такие как OpenGL для графики или HTML5 для веб-приложений, реже сталкиваются с проблемами совместимости.

Эти методы помогают выявить проблемы до их появления у конечного пользователя, что делает ПО более надёжным и функциональным.

Практическое задание:

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите чтение с компакт дисков.

**Билет 14 МДК04.01**

**Теоретический вопрос:**

**Чистая загрузка операционной системы. Что представляет собой чистая загрузка ОС, в каких случаях она используется, и как выполняется этот процесс в системах Windows, Linux и Mac OS? Объясните основные шаги для каждой системы и приведите примеры практического применения.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Чистая загрузка операционной системы — это процесс загрузки с минимальным набором драйверов, служб и приложений. Она используется для диагностики и устранения проблем, связанных с конфликтами драйверов, приложений или системных служб. Чистая загрузка помогает изолировать источник проблемы, временно отключая все ненужные компоненты.

**Выполнение чистой загрузки в Windows:**

1. Откройте конфигурацию системы (Win+R → msconfig).
2. На вкладке **"Службы"** установите флажок **"Не отображать службы Microsoft"** и отключите все сторонние службы.
3. На вкладке **"Автозагрузка"** в "Диспетчере задач" отключите программы автозагрузки.
4. Перезагрузите систему.

**Пример:**

После установки стороннего антивируса Windows стала загружаться медленно. Выполнив чистую загрузку, администратор отключил антивирус, что восстановило нормальную скорость загрузки.

**Выполнение чистой загрузки в Linux:**

1. Отключите ненужные службы с помощью команды:

sudo systemctl disable apache2 (это мы web сервер отрубили)

1. Используйте меню GRUB для добавления параметров single или recovery, чтобы загрузиться в однопользовательском режиме.
2. Внесите изменения для устранения конфликта (например, удалите проблемный драйвер).

**Пример:**

После установки драйвера видеокарты Linux зависал при запуске. Однопользовательский режим позволил удалить драйвер, чтобы восстановить систему.

**Выполнение чистой загрузки в Mac OS:**

1. Перезагрузите Mac и удерживайте клавишу Shift при включении.
2. Войдите в безопасный режим (загрузка минимальных драйверов и служб).
3. Устраните проблему, отключив автозапуск программ через **"Системные настройки → Пользователи и группы → Объекты входа"**.

**Пример:**

Программа для обработки видео вызывала сбои при запуске Mac. В безопасном режиме её автозапуск был отключён, что устранило проблему. Чистая загрузка является универсальным методом диагностики, который позволяет выявлять конфликты и повышать стабильность работы системы.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите построение графиков в приложении калькулятор.

**Билет 15 МДК04.01**

**Теоретический вопрос:**

**Ситуации, требующие чистой загрузки. В каких случаях возникает необходимость выполнения чистой загрузки операционной системы, и как этот процесс помогает решать проблемы? Приведите примеры из реальных сценариев, связанных с драйверами, автозагрузкой программ и конфликтами приложений.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Чистая загрузка необходима в ситуациях, когда возникают проблемы с загрузкой системы или её работой из-за конфликтов между драйверами, программами или системными службами. Выполнение чистой загрузки позволяет изолировать проблемный компонент, временно отключив все ненужные элементы.

**Сценарии применения:**

1. **Медленная загрузка системы:** Часто возникает из-за множества программ в автозагрузке или конфликтов между ними. **Пример:** После установки программы для мониторинга ресурсов Windows стала загружаться медленно. Чистая загрузка отключила эту программу, что восстановило нормальную скорость загрузки.
2. **Конфликты драйверов:** Возникают при установке новых устройств или обновлении драйверов. **Пример:** После установки новой видеокарты Linux начал зависать. Чистая загрузка в однопользовательском режиме помогла отключить драйвер, вызвавший конфликт.
3. **Конфликты приложений:** Если два приложения используют один и тот же системный ресурс, это может привести к сбоям. **Пример:** На Mac OS приложение для редактирования видео конфликтовало с другой программой, использующей графический процессор. В безопасном режиме удалось отключить конфликтующее приложение.
4. **Системные сбои после обновлений:** Обновления могут привести к несовместимости драйверов или приложений. **Пример:** После обновления Windows система перестала загружаться. Чистая загрузка помогла выявить драйвер, несовместимый с новой версией ОС.

Чистая загрузка является важным инструментом для диагностики и устранения проблем, позволяя изолировать источник сбоя и минимизировать риск потери данных.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите доступ к магазину приложений Store.

**Билет 16 МДК04.01**

**Теоретический вопрос:**

**Роль динамически загружаемых библиотек (DLL) в совместимости программного обеспечения. Что такое DLL, какие преимущества они предоставляют, и как их отсутствие или неправильная версия могут вызвать проблемы совместимости? Объясните на примерах работы системных библиотек и ошибок, связанных с их отсутствием.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Динамически загружаемые библиотеки (DLL) — это файлы, содержащие код и данные, которые загружаются в память приложения по мере необходимости. DLL предоставляют приложениям возможность повторно использовать код и экономить системные ресурсы.

**Преимущества использования DLL:**

1. **Экономия памяти:** Код библиотеки загружается только при необходимости.
2. **Повторное использование кода:** Разные приложения могут использовать одну и ту же библиотеку.
3. **Обновляемость:** Обновление библиотеки позволяет внести изменения, не затрагивая исходный код приложения.

**Пример 1: Использование системных библиотек Windows.** Системные DLL, такие как kernel32.dll, используются для выполнения базовых операций, таких как работа с файлами. Обновление ОС может изменить их версии, что вызовет проблемы с работой приложений.

**Пример 2: Ошибка из-за отсутствия библиотеки.** Приложение, разработанное на C++, может выдать ошибку из-за отсутствия MSVCP140.dll, входящей в Microsoft Visual C++ Redistributable. Решение заключается в установке этого пакета, чтобы приложение могло корректно работать.

DLL играют ключевую роль в совместимости, но требуют точного соответствия версий и наличия необходимых компонентов для обеспечения стабильности приложений.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, отключите автозапуск для всех устройств.

**Билет 17 МДК04.01**

**Теоретический вопрос:**

**Анализ проблем совместимости, связанных с библиотеками DLL. Какие инструменты помогают выявить проблемы совместимости с динамическими библиотеками, как они работают, и как их использование позволяет устранять ошибки? Приведите примеры использования Dependency Walker и команды ldd.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Проблемы совместимости часто связаны с отсутствием или неправильными версиями библиотек. Для их анализа используются специальные инструменты, которые помогают определить, какие зависимости приложения не удовлетворены.

**Инструменты для анализа:**

1. **Dependency Walker (Windows):** Это программа, которая отображает список всех библиотек, используемых приложением, и указывает, какие из них отсутствуют или несовместимы. **Пример:** Если приложение на Windows не запускается, Dependency Walker позволяет увидеть, что отсутствует библиотека MSVCR120.dll. Установка соответствующего пакета Visual C++ Redistributable решает проблему.
2. **ldd (Linux):** Команда ldd показывает все зависимости библиотеки или программы, а также отсутствующие компоненты. **Пример:** При запуске программы на Linux ошибка "segmentation fault" может быть вызвана отсутствием libssl.so.1.1. Использование команды ldd указывает на эту проблему, что позволяет установить нужную библиотеку через менеджер пакетов.

**Устранение ошибок:**

1. Обновление или установка библиотек.
2. Использование контейнеров, таких как Docker, для создания изолированных окружений.
3. Переключение на статическую компоновку, чтобы исключить зависимости от версий библиотек.

Использование этих инструментов позволяет эффективно выявлять и устранять проблемы совместимости, связанные с библиотеками, обеспечивая стабильность работы приложений.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите установку приложений со съёмных носитилей.

**Билет 18 МДК04.01**

**Теоретический вопрос:**

**Учет аппаратных компонентов в организации. Какие задачи решает учет аппаратных компонентов, как он помогает оптимизировать работу оборудования, и какие инструменты используются для учета? Объясните это на примерах систем мониторинга и инвентаризации.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Учет аппаратных компонентов (Hardware Asset Management, HAM) — это процесс управления всеми аппаратными устройствами в организации. Он включает в себя инвентаризацию, мониторинг состояния и планирование обновлений оборудования. Учет помогает предотвратить сбои, сократить затраты и обеспечить надёжность инфраструктуры.

**Основные задачи учета:**

1. **Контроль состояния оборудования:** Регулярный мониторинг позволяет своевременно выявлять неисправности, такие как перегрев серверов или снижение производительности жёстких дисков. **Пример:** В дата-центре системы мониторинга, такие как Zabbix, оповещают администратора о перегреве процессора, что позволяет предотвратить поломку.
2. **Планирование обновлений:** Системы учета помогают определить, какие устройства устарели и требуют замены. **Пример:** Система AssetExplorer выявила старые рабочие станции, которые не поддерживают новое программное обеспечение, что позволило своевременно запланировать обновления.
3. **Оптимизация затрат:** Учет помогает избежать избыточных расходов на оборудование и оптимально распределять ресурсы. **Пример:** Использование системы Spiceworks позволяет отслеживать, какие устройства не используются, и перераспределить их.

**Инструменты учета:**

1. **Встроенные утилиты:**
   * **Windows:** Диспетчер устройств предоставляет данные о состоянии оборудования.
   * **Linux:** Команда lshw показывает характеристики аппаратных компонентов.
2. **Системы мониторинга:**
   * **Zabbix:** Позволяет отслеживать производительность серверов и сетевых устройств.
   * **Nagios:** Помогает контролировать температуру и состояние жёстких дисков.
3. **Программное обеспечение ITAM:**
   * **Spiceworks:** Бесплатное ПО для автоматического учёта активов.
   * **ManageEngine AssetExplorer:** Платформа для управления жизненным циклом оборудования.

Учет аппаратных компонентов обеспечивает стабильность работы инфраструктуры и помогает оптимизировать управление активами.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите доступ к центру обновления Windows.

**Билет 19 МДК04.01**

**Теоретический вопрос:**

**Мониторинг аппаратных компонентов. Какие системы используются для мониторинга оборудования, как они помогают предотвращать сбои, и чем отличаются от встроенных утилит? Приведите примеры использования Zabbix и Nagios.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Мониторинг аппаратных компонентов — это процесс контроля состояния оборудования в реальном времени. Он помогает выявлять проблемы на ранних стадиях и предотвращать сбои, что особенно важно для серверов и критически важных устройств.

**Основные системы мониторинга:**

1. **Zabbix:**
   * Предоставляет данные о загруженности процессоров, использовании памяти, состоянии дисков.
   * Настраивает автоматические оповещения об аномалиях. **Пример использования:** В крупной компании Zabbix обнаружил перегрузку сервера, что позволило перераспределить ресурсы и избежать сбоя.
2. **Nagios:**
   * Следит за температурой процессоров, состоянием жёстких дисков, доступностью сетевых интерфейсов. **Пример использования:** Nagios предупредил администратора о высоком уровне ошибок на сетевом интерфейсе, что позволило заменить неисправное оборудование до возникновения проблем.

**Преимущества перед встроенными утилитами:**

1. **Реальное время:** Системы мониторинга предоставляют актуальные данные в режиме реального времени, тогда как встроенные утилиты, такие как Диспетчер устройств Windows или lshw в Linux, требуют ручного запуска.
2. **Оповещения:** Интеграция с системами уведомлений позволяет администраторам быстро реагировать на сбои.
3. **Широкий функционал:** Системы мониторинга могут контролировать не только оборудование, но и сетевую инфраструктуру.

Мониторинг позволяет автоматизировать управление инфраструктурой, минимизируя простои и снижая затраты на обслуживание.

Практическое задание:

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, добавьте в меню Пуск команду Выполнить

**Билет 20 МДК04.01**

**Теоретический вопрос:**

**Диагностика аппаратных неисправностей. Какие основные этапы диагностики используются для выявления проблем аппаратного обеспечения компьютера, и как правильно идентифицировать неисправность? Расскажите о применении визуальной проверки, звуковых сигналов BIOS и профессиональных инструментов диагностики.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Диагностика аппаратных неисправностей включает несколько ключевых этапов, которые позволяют определить источник проблемы. Это особенно важно при сбоях компьютера, таких как невозможность включения или нестабильная работа.

**Основные этапы диагностики:**

1. **Визуальная проверка:**
   * Проверьте наличие характерного запаха гари, вздутых конденсаторов или других визуальных признаков повреждений.
   * Убедитесь, что кабели и компоненты подключены правильно.

**Пример:** Если блок питания не работает, вентиляторы не крутятся, а винчестер не издает звуков, скорее всего, проблема связана с блоком питания.

1. **Звуковые сигналы BIOS:**

BIOS передаёт звуковые сигналы для диагностики неисправностей:

* + **1 длинный, 3 коротких:** проблема с видеокартой.
  + **Повторяющиеся короткие сигналы:** перегрев процессора или проблемы с напряжением.

**Пример:** Если BIOS выдаёт 3 коротких сигнала, это может указывать на проблему с оперативной памятью.

1. **Применение профессиональных инструментов диагностики:**

Используйте диагностические карты (например, ДП-1), которые отображают код ошибки в шестнадцатеричном виде. Это особенно полезно, если процессор исправен, но проблема в других компонентах.

**Пример:** Диагностическая карта указала код ошибки, связанный с неисправностью чипсета.

Эти этапы диагностики помогают эффективно выявить и устранить аппаратные неисправности, обеспечивая стабильную работу системы.

Практическое задание:

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите пользователям перемещать панель задач.

**Билет 21 МДК04.01**

**Теоретический вопрос:**

**Программно-аппаратная диагностика. Какие методы программной диагностики используются для проверки состояния аппаратного обеспечения компьютера, и как они помогают выявить потенциальные проблемы? Объясните на примерах использования Memtest86 и анализа SMART-атрибутов HDD.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Программная диагностика позволяет оценить состояние аппаратных компонентов компьютера при его включении. Это особенно важно для проверки оперативной памяти, жёсткого диска и процессора.

**Основные методы программной диагностики:**

1. **Тестирование оперативной памяти:**

Используйте программы, такие как Memtest86, которые работают под MS-DOS. Они проверяют каждый модуль памяти на наличие ошибок.

**Пример:** Если Memtest86 выявляет ошибки в одном из банков оперативной памяти, модуль требует замены.

1. **Анализ SMART-атрибутов жёсткого диска:**

Системы SMART предоставляют данные о состоянии HDD, включая количество переназначенных секторов, время работы и скорость вращения шпинделя.

**Пример:** Программа Active SMART указала на рост числа секторов с ошибками, что является признаком износа диска.

1. **Burn-in тесты:**

Эти тесты проверяют стабильность работы оборудования под высокой нагрузкой. Рекомендуется проводить их не менее 24 часов.

**Пример:** После сборки нового компьютера burn-in тест выявил перегрев процессора из-за неправильно установленного кулера.

1. **Проверка температуры:**

Регулярное измерение температуры процессора, видеокарты и жёсткого диска помогает предотвратить перегрев.

**Пример:** Программа HWMonitor показала высокую температуру видеокарты, что указывает на необходимость чистки системы охлаждения.

Программная диагностика является незаменимым инструментом для выявления скрытых проблем аппаратного обеспечения и продления срока службы компьютера.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, запретите операционной системе и установленным программам создавать и отображать ярлыки для недавно открывавшихся документов.

**Билет 22 МДК04.01**

**Теоретический вопрос:**

**Жизненный цикл программного обеспечения. Какие процессы входят в жизненный цикл ПО согласно ГОСТ Р ИСО/МЭК 12207-2010, и как каждый из них обеспечивает качество программного продукта?**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Жизненный цикл программного обеспечения описывает полный набор процессов, необходимых для создания, внедрения, эксплуатации и завершения использования программного продукта. Согласно ГОСТ Р ИСО/МЭК 12207-2010, процессы делятся на следующие категории:

1. **Процессы соглашений:**

Регулируют взаимодействие между сторонами (заказчиком и исполнителем) для достижения согласованных целей.

**Пример:** Подписание контракта на разработку ПО с чётко определёнными требованиями и условиями сдачи продукта.

1. **Процессы организационного обеспечения:**

Включают управление ресурсами, инфраструктурой и компетенциями, необходимыми для выполнения проекта.

**Пример:** Организация команды разработчиков с распределением ролей, настройка серверов для тестирования.

1. **Технические процессы:**

Охватывают проектирование, разработку, тестирование и сопровождение. Эти процессы детализированы в самом стандарте.

**Пример:** Проектирование архитектуры клиент-серверного приложения.

1. **Процессы сопровождения:**

Обеспечивают адаптацию ПО к изменяющимся условиям.

**Пример:** Выпуск обновлений для совместимости с новыми версиями операционных систем.

1. **Процессы завершения:**

Включают изъятие ПО из обращения, утилизацию данных и ресурсов.

**Пример:** Завершение использования корпоративной системы с переносом данных в новую платформу.

Эти процессы обеспечивают системный подход к разработке и эксплуатации ПО, минимизируя риски и увеличивая надёжность продукта.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите команду «Завершение сеанса» из меню «Пуск» и запретить пользователям ее восстановление.

**Билет 23 МДК04.01**

**Теоретический вопрос:**

**Технические процессы. Как технические процессы стандарта ГОСТ Р ИСО/МЭК 12207-2010 поддерживают соответствие программного обеспечения требованиям пользователей? Раскройте роль проектирования, реализации и тестирования.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Технические процессы представляют собой основу жизненного цикла разработки ПО. Они обеспечивают выполнение требований заказчиков и соответствие программного продукта установленным стандартам.

1. **Определение требований правообладателей:**

Включает сбор, анализ и согласование ожиданий всех заинтересованных сторон.

**Пример:** В системе электронного документооборота учитываются требования по защите данных и удобству интерфейса.

1. **Проектирование архитектуры системы:**

Разрабатывается структура системы, которая включает модули, интерфейсы и взаимодействия между ними.

**Пример:** Архитектура распределённой системы предусматривает взаимодействие через API.

1. **Реализация:**

Включает написание кода, тестирование модулей и их интеграцию.

**Пример:** При реализации веб-приложения используется фреймворк Django для ускорения разработки.

1. **Комплексирование и тестирование:**

Интеграция компонентов и проверка их совместимости.

**Пример:** Системное тестирование ERP-системы выявило несовместимость между модулями отчётности и складского учёта.

Эти процессы обеспечивают качество и стабильность программного продукта на всех этапах жизненного цикла.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите значок «Документы» из меню «Пуск» и его подменю.

**Билет 24 МДК04.01**

**Теоретический вопрос:**

**Сопровождение программного обеспечения. Почему процесс сопровождения ПО является ключевым этапом жизненного цикла, и какие задачи он решает?**

**+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**Ответ:**

Сопровождение программного обеспечения включает действия по поддержанию и улучшению программного продукта на протяжении всего срока его эксплуатации. Это один из самых продолжительных этапов жизненного цикла.

**Основные задачи процесса сопровождения:**

1. **Исправление ошибок:**

Устраняются дефекты, которые возникают при эксплуатации ПО.

**Пример:** В онлайн-банкинге устранена уязвимость, позволявшая обойти двухфакторную аутентификацию.

1. **Обновление функциональности:**

ПО адаптируется к изменяющимся потребностям пользователей и новым требованиям.

**Пример:** Включение возможности оплаты через криптовалюту в интернет-магазине.

1. **Поддержка совместимости:**

Обеспечивается корректная работа с новыми версиями операционных систем и оборудования.

**Пример:** Обновление драйверов для работы со сканерами на Windows 11.

1. **Оптимизация производительности:**

Улучшается быстродействие системы и уменьшается её нагрузка на ресурсы.

**Пример:** Оптимизация алгоритмов обработки данных в системе Big Data.

Процесс сопровождения гарантирует, что программное обеспечение остаётся актуальным, надёжным и полезным для пользователей.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите значок «Изображения» из меню «Пуск».

**Билет 25 МДК04.01**

**Теоретический вопрос:**

**Изъятие программного обеспечения из обращения. Какие этапы включает процесс изъятия программного обеспечения, и как этот процесс влияет на безопасность данных и инфраструктуры? Раскройте основные задачи и приведите примеры действий при завершении эксплуатации систем.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Ответ:**

Изъятие программного обеспечения из обращения — это процесс завершения использования программного продукта, включающий безопасное прекращение эксплуатации, утилизацию данных и ресурсов, а также подготовку системы к переходу на новое решение. Этот процесс является специальным случаем, описанным в ГОСТ Р ИСО/МЭК 12207-2010, и направлен на минимизацию рисков, связанных с безопасностью данных и функциональностью инфраструктуры.

**Основные этапы изъятия ПО:**

1. **Оценка текущего состояния системы:**

Анализируется необходимость продолжения эксплуатации системы, её эффективность и соответствие текущим требованиям.

**Пример:** Старая CRM-система больше не соответствует бизнес-требованиям компании и подлежит замене.

1. **Перенос данных:**

Включает экспорт данных в совместимый формат и их интеграцию в новое программное обеспечение.

**Пример:** Данные из старой бухгалтерской программы, такой как **1С:Бухгалтерия 8**, переносятся в новую версию **1С:ERP**, чтобы улучшить функционал и соответствовать современным бизнес-требованиям. При этом старые базы данных архивируются, а носители с конфиденциальной информацией утилизируются.

1. **Обеспечение безопасности:**

Гарантируется, что конфиденциальные данные не могут быть восстановлены, а доступ к системе для несанкционированных пользователей закрыт. Используются методы шифрования, перезаписи или физического уничтожения.

1. **Утилизация оборудования:**

Носители информации утилизируются или перераспределяются внутри организации.

**Пример:** Жёсткие диски уничтожаются в соответствии с регламентами ФСТЭК, а серверы используются для других задач.

1. **Документирование процесса:**

Включает составление отчётов о выполненных действиях, включая перечень перенесённых данных и результаты удаления.

**Заключение:**

Процесс изъятия программного обеспечения из обращения включает комплекс мероприятий, направленных на безопасное завершение использования, сохранение целостности данных и эффективное перераспределение ресурсов.

**Практическое задание:**

Создайте пользователя EXAM\_НОМЕР\_БИЛЕТА.

Используя групповые политики, удалите команду «Выполнить» из меню «Пуск», браузера Internet Explorer и диспетчера задач.