# Над объектом какого типа выполняются все графические операции?

Все графические операции в Java выполняются над объектами типа Graphics. Этот класс предоставляет методы для рисования на компонентах. Когда вы переопределяете метод paint в вашем компоненте, вы получаете объект Graphics в качестве параметра, который вы можете использовать для рисования фигур, текстов и изображений.

# Почему рекомендуется использовать ограниченный набор цветов?

Использование ограниченного набора цветов может улучшить производительность. Так же при использовании ограниченного набора цветов, проще вносить изменения и обновления в дизайн, так как не нужно отслеживать множество различных цветовых значений.

# Почему рекомендуется использовать ограниченный набор шрифтов?

Ограниченный набор шрифтов позволяет делать текст более гармоничным и легким для восприятия. Так же ограничение числа шрифтов может улучшить производительность и ускорить загрузку страниц. С меньшим количеством шрифтов проще вносить изменения и обновления в дизайн. Это упрощает процесс редактирования и поддержания согласованности в будущем.

# Как получить метрики шрифтов?

Чтобы получить метрики шрифтов в Java апплетах, вы можете использовать класс Font и его методы. Чтобы получить массив установленных шрифтов можно воспользоваться конструкцией GraphicsEnvironment.getLocalGraphicsEnvironment().getAllFonts()

# Когда возникают события?

События возникают в результате различных взаимодействий и изменений состояния, связанных с пользовательским интерфейсом и внутренними процессами апплета.

Примеры:

1. Клики и перемещения мыши
2. Клавиатурные события
3. Изменение размеров окна апплета

# Какой метод какого класса получает управление при возникновении события? Что передается ему в качестве параметра?

При возникновении события управление передается методу, который зарегистрирован для обработки этого события. Пример: Кнопка и метод для обработки события нажатия кнопки. В метод отвечающий за обработку события в качестве параметра передается объект типа Event

# Какие методы отвечают за обработку простых событий от мыши?

1. mouseClicked - Вызывается, когда пользователь нажимает и отпускает кнопку мыши на компоненте.
2. mousePressed - Вызывается, когда пользователь нажимает кнопку мыши на компоненте.
3. mouseReleased -Вызывается, когда пользователь отпускает кнопку мыши на компоненте.
4. mouseEntered - Вызывается, когда курсор мыши входит в границы компонента.
5. mouseExited - Вызывается, когда курсор мыши покидает границы компонента.

# Какое значение (true или false) должен вернуть метод-обработчик для того, чтобы событие считалось обработанным и не передавалось на дальнейшую обработку методу суперкласса?

Метод-обработчик события должен вернуть true, чтобы сигнализировать, что событие было обработано и не должно передаваться на дальнейшую обработку. Если метод возвращает false, это означает, что событие не было полностью обработано, и оно может быть передано другим обработчикам или методу суперкласса.

# Какие методы отвечают за обработку простых событий от клавиатуры?

1. keyPressed - Вызывается, когда пользователь нажимает клавишу на клавиатуре.
2. keyReleased - Вызывается, когда пользователь отпускает клавишу на клавиатуре.
3. keyTyped - Вызывается, когда пользователь нажимает клавишу, которая печатает символ (например, буквы и цифры).

# Для чего нужен фокус ввода?

Фокус ввода — это состояние, при котором компонент графического интерфейса может получать ввод от пользователя, например, с клавиатуры. Компонент, имеющий фокус, будет обрабатывать события клавиатуры, такие как нажатия клавиш, в то время как другие компоненты не будут реагировать на эти события.