

## GUI (graphical user interface)

GUI — это графический пользовательский интерфейс. Название происходит от английского словосочетания graphical user interface. Так называется внешний вид программы — то, что видит пользователь и с чем он может взаимодействовать. Это, например, компоненты графического интерфейса: окна, кнопки и так далее.

Графический интерфейс есть в большинстве современных программ, сайтов и сервисов. «Общаться» с программой таким образом куда проще, чем отдавать ей команды через консоль или писать скрипты. Вся работа с компьютером становится наглядной и понятной для человека.

У операционных систем тоже есть GUI. Панель задач, окна, папки, курсор — все это элементы графического интерфейса. К нему существуют определенные требования, чтобы внешний вид программы был понятным и удобным для человека.

Название по-русски произносится как ГУИ. Иногда в материалах по информатике используют адаптированную аббревиатуру ГИП — графический интерфейс пользователя.

## Для чего нужен GUI

Графическими интерфейсами пользуются абсолютно все, у кого есть компьютер, смартфон или планшет. GUI есть в большинстве современных программ, кроме тех, которые работают «в фоне» и не взаимодействуют с пользователем.

GUI нужен, чтобы пользователям было удобнее. Альтернатива графическому интерфейсу — консольный интерфейс CLI, то есть программа для командной строки. У нее нет кнопок и окошек, а чтобы управлять программой, нужно писать команды в консоль. Такой подход бывает удобным для программистов, но только в случае специализированного профессионального ПО. А программами и приложениями для личных дел проще и удобнее пользоваться через GUI.

GUI нужен, чтобы опыт пользователя при работе с программой был максимально комфортным. А еще — очевидным: человек при взгляде на интерфейс должен понимать, что значит каждый его компонент.

## Какие компоненты есть в GUI

**Окна.** Окно программы — это один из базовых элементов графического интерфейса. Стандартный пример — окно браузера или «Проводника» в операционной системе. Окно можно увеличить, уменьшить, свернуть, развернуть и закрыть. В верхней его части нередко располагают меню.

**Меню.** Это панель, которая дает доступ к основным возможностям приложения или сайта. В меню программы можно, например, открыть или закрыть файл, настроить разные параметры, создать что-то или отредактировать, вставить контент или сделать что-то еще — зависит от программы. Меню сайта — это обычно ссылки на полезные страницы ресурса. А, например, меню игры — сохранение, загрузка, параметры графики и так далее.

**Кнопки и интерактивные элементы.** Кнопки — это участки, на которые можно нажимать мышью, чтобы выполнить то или иное действие. Но это не единственный интерактивный элемент в пользовательском интерфейсе. Есть пиктограммы — аналоги кнопок, на которых изображены картинки. Есть интерактивные области: что-то происходит, когда пользователь проводит по ним мышью или нажимает. Есть поля для ввода текста и окошки для отображения информации. Все это интерактивные элементы.

**Указатели.** Информатика определяет указатель как часть графического интерфейса. Это курсор мыши в виде стрелочки или, например, текстовый курсор — вертикальная палочка, мигающая в поле ввода. Человек манипулирует указателем с помощью мыши, тачпада или трекбола, нажимает на кнопки, взаимодействует с интерактивными элементами.

Обычно указатель — это часть графического интерфейса операционной системы. То есть сторонние программы не создают свои указатели, а пользуются тем, что уже есть в ОС. То же самое касается сайтов.

**Манипуляции.** Действия, которые можно совершить с интерфейсом, тоже считаются его частью. Например, на кнопку можно нажать, меню — открыть и закрыть, окно — расширить. Еще есть drag-n-drop и другие способы взаимодействия с интерактивными элементами. Все это делается с помощью мыши или клавиатуры, иногда — других устройств ввода и вывода.

Когда программист создает графический интерфейс, он прописывает, как его компоненты будут реагировать на то или иное действие пользователя. А

самую возможность совершать эти действия дает операционная система, а также устройства ввода-вывода у человека.

## Каким может быть GUI

- Стандартный графический интерфейс — это окно, в котором есть разные пиктограммки, кнопки, поля и так далее. Пример — окно «Проводника» в операционной системе. Интерфейс понятный, наглядный и функциональный. Так называют GUI, которые реализованы средствами самой ОС.
- Существуют также истинно графические интерфейсы — так называют красивые GUI, которые реализованы с помощью сторонних инструментов. В них могут быть необычные, нестандартные компоненты, например, кнопки разных форм и цветов, картинка-кнопки, анимации и так далее. Пример — сайты: там интерфейсы не зависят от операционной системы пользователя и отрисовываются в браузере. А придумывает их команда программистов и дизайнеров, с нуля. Средства ОС там используются очень ограниченно.
- Трёхмерные графические интерфейсы в основном встречаются в играх и в других программах, которые подразумевают работу с 3D-моделями. Например, рабочая область 3D-редактора будет иметь трёхмерный интерфейс с тенями, сеткой и отображением объекта, который сейчас рисуют.

## Требования к GUI

**Наглядность.** С графическими интерфейсами работает принцип DWIM — Do What I Mean, то есть «Делай то, что я имею в виду». Этот принцип означает, что интерфейс должен быть интуитивно ясным, а пользователь легко мог понять, что делает тот или иной элемент. Можно сказать, что интерфейс должен быть предсказуемым: например, кнопка «Сохранить» должна сохранять файл как есть, а не экспортировать в другой формат.

**Понятность.** Ещё один важный принцип — понятность для тех, кто впервые работает с программой. Это не совсем то же самое, что наглядность и предсказуемость. Нужно, чтобы человек мог легко разобраться, как пользоваться самими элементами интерфейса. Пользователь должен понимать, где что можно найти, куда нажимать и так далее. Например, важная кнопка с непонятной подписью или там, где её никто не найдет, — пример не очень хорошего интерфейса.

**Удобство.** Пользоваться графическим интерфейсом должно быть удобно — это то, ради чего он в принципе создавался. То есть важные функции должны находиться под рукой, а неважные — подальше. В хороших интерфейсах бывает возможность настраивать отображение инструментов и элементов. Если речь о сайтах, то они должны подстраиваться под устройство и экран пользователя (так называемая адаптивная верстка).

Удобство — очень широкое понятие. Для его измерения даже есть специальные метрики: по ним определяют, насколько быстро и полно человек может разобраться в интерфейсе.

**Эффективность.** Эффективный интерфейс — это такой, который помогает достичь нужного результата как можно быстрее. Сюда можно отнести два правила:

- интерфейс не должен «тормозить» и зависать;
- интерфейс должен давать возможность быстро добраться до нужной функции.

На благо эффективности, например, работают горячие клавиши — они помогают быстро выполнить нужное действие без захода в меню. И наоборот: если для каждого простого действия надо много кликать и долго ждать, это не самый хороший интерфейс. Особенно строго с этим в веб-разработке, где считается, что каждый лишний клик отдаляет пользователя от цели — он скорее уйдет с сайта, чем доберется до нужного места.

**Обратная связь.** Выше мы говорили, что интерфейс не должен «зависать». Тут есть еще один момент: обратная связь. Графический интерфейс должен быть отзывчивым, показывать пользователю, что происходит в данный момент.

Если программа загружается — пусть она покажет экран загрузки, а не просто замолчит. Иначе пользователь просто не поймет, что происходит, и может решить, что все сломалось. То же самое касается ошибок, потери сети, подгрузки каких-то сложных функций — всего, что может потребовать времени.

Даже простые интерактивные элементы нужно создавать такими, чтобы при наведении или клике на них они меняли внешний вид. Так пользователь получит обратную связь: программа увидела его действие. Пример — кнопка, которая темнеет, если навести на нее курсор, и подсвечивается в момент нажатия.

**Последовательность.** Последовательность означает, что элементы должны быть расположены в правильном порядке — таком, который удобен пользователю. Тут хороший пример — интерфейс операционной системы. Люди уже привыкли, что кнопка «Пуск» находится слева, а часы — справа, и если изменить это расположение элементов по умолчанию, пользователи запутаются. Поэтому даже другие операционные системы теперь располагают часы справа, а кнопку меню системы — слева.

**Красота.** Наконец, интерфейс должен быть привлекательным: иметь приятную цветовую гамму, грамотно продуманные очертания. Это не означает, что каждый GUI должен быть произведением искусства, просто не стоит делать его аляповатым и отталкивающим. У эстетике есть свои правила, и ими пользуются дизайнеры при проектировании графических интерфейсов.

### Примеры GUI

- Графический интерфейс операционной системы: рабочий стол, панель задач и меню «Пуск», трей, иконки, окна и так далее.
- Интерфейс браузера: окно, список открытых вкладок, панель закладок сверху, адресная строка, кнопки «Вперед», «Назад» и другие, меню, аккаунт, список расширений.
- Графический интерфейс сайта: шапка и подвал, меню, основной контент страницы, кнопки и формы, виджеты и прочие элементы.
- Интерфейс среды разработки: окно программы, рабочая область для ввода кода, меню управления проектом и программой, меню запуска и отладки, дополнительные расширения, контроль версий и прочее. В масштабных средах может быть очень много функций.
- Интерфейс 3D-редактора: «сцена», где расположены объект, свет и текстуры — это, кстати, пример трехмерного интерфейса. Также в нем есть меню для работы с объектами, различные инструменты, горячие клавиши и прочее.

### Преимущества GUI

- **Удобство.** Пользоваться GUI намного проще и понятнее, чем командной строкой. Графические интерфейсы в принципе сделаны так, чтобы в них мог разобраться любой.

- **Широкая аудитория.** Приложением с графическим интерфейсом может воспользоваться кто угодно, даже человек, который не слишком хорошо разбирается в компьютерах.
- **Понятность.** Функции, которые есть в графическом интерфейсе, видны наглядно: человеку не приходится читать мануалы, чтобы узнать команды для управления программой.
- **Скорость работы.** Зачастую нажать кнопку в GUI куда быстрее, чем писать строки кода для консоли. А еще в графическом интерфейсе можно быстрее разобраться.

## Недостатки GUI

Графические интерфейсы менее производительны, чем текстовые и CLI. Ведь программа также должна отрисовывать графику и соотнести графические компоненты с определенными действиями. А в некоторых случаях прописать что-то в консоли оказывается быстрее и легче: например, если нужно скопировать много файлов или запустить поиск по определенной фразе во всех текстовых документах.

Профессиональное ПО для разработчиков нередко не имеет GUI, потому что он будет просто излишним. Например, популярные утилиты для работы с кодом — консольные: у них текстовый интерфейс, и программист взаимодействует с ним через командную строку.

## Как создают GUI

Создание графического интерфейса можно условно разделить на несколько этапов.

Сначала создатели проекта продумывают концепт и функциональность. Потом дизайнеры на основании требований рисуют, как будет выглядеть программа. Работа дизайнеров — создать изображение интерфейса и его архитектуру, а не реализацию. Они должны учесть даже мелкие детали, например как будет меняться цвет кнопки при нажатии.

Разработчики воплощают в жизнь интерфейс, который нарисовали дизайнеры. Они программируют элементы, связывают их с определенными действиями, пишут функции. Как именно это делается, зависит от сферы:

- в веб-разработке графические и функциональные элементы создают с помощью HTML и CSS, какие-то сложные компоненты программируют на JavaScript;

- для приложений под определенные ОС есть среды разработки с возможностью создавать графические интерфейсы, например Visual Studio. А есть библиотеки компонентов, в которых уже собраны готовые наборы элементов.