



Лекция № 7

**Часть I. Машинная Графика
Графическая Система
Основные концепции ЯГС (GKS)
(продолжение)**

Основные концепции (GKS). Концепция Ввода.

Мир машинной графики еще более обогатился, когда графическая система смогла не только создавать изображения на носителе изображения устройства вывода, но и обрабатывать данные, которые вводит оператор на станции.

Эта возможность - диалог, давший толчок быстрому росту использования устройств и систем машинной графики. Действия по указанию, выбору, рисованию, размещению или стиранию, производимые человеком непосредственным образом, и постоянный ответ системы на эти действия составляют действительно естественный способ взаимодействия человека с машинной средой.

Именно для обеспечения такого взаимодействия и предназначена **интерактивная машинная графика**. Это самый мощный инструмент, подчиняющий человеко-машинный интерфейс требованиям человека.

В состав ЯГС входят функции ввода величин, поступающих с устройств различного класса, работающих в различных режимах.

Чтобы использовать широкий спектр оборудования на уровне, не зависящем от устройств, **характеристики физических устройств ввода различного типа отображаются на логические устройства ввода**. Модель ЯГС описывает процесс ввода в терминах логических и физических устройств и их взаимного отображения.

Логическое устройство предоставляет прикладной программе значение логического ввода. Тип этих данных определяется классом ввода.

Логическое устройство ввода является обобщением одного или более физических устройств и служит для передачи в программу значений логического ввода.

Значение логического ввода - значение, связанное с логическим устройством и преобразованием передаваемых данных.

Класс ввода - логически эквивалентный по выполняемым функциям набор устройств ввода.

Основные концепции (GKS). Концепция Ввода. Классы ввода.

ВВОД ПОЗИЦИИ (LOCATOR) - предоставляет прикладной программе координаты точки (позицию) в мировых координатах.

ВВОД ПОСЛЕДОВАТЕЛЬНОСТИ ПОЗИЦИЙ (STROKE) - предоставляет прикладной программе последовательность позиций в мировых координатах.

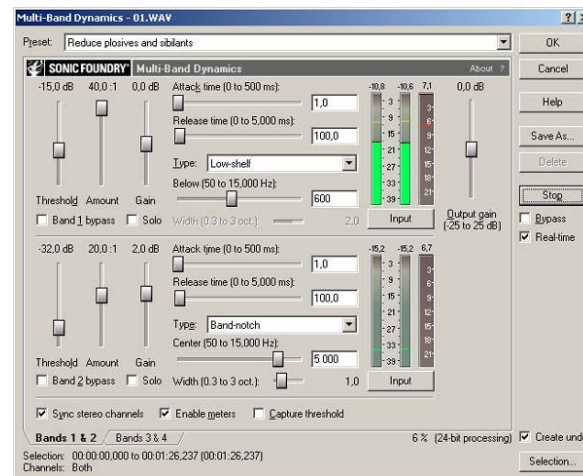
ВВОД ЧИСЛА (VALUATOR) предоставляет прикладной программе вещественное число.

ВЫБОР альтернативы (CHOICE) предоставляет прикладной программе неотрицательное целое число, соответствующее выбору из некоторого числа возможностей.

УКАЗАНИЕ объекта (PICK) предоставляет прикладной программе имя сегмента и идентификатор указания.

ВВОД СТРОКИ (STRING) предоставляет прикладной программе последовательность литер;

Из перечисленных классов логических устройств ввода чисто графическими можно считать только ввод позиции, ввод последовательности позиции и указание объекта. Тем не менее можно с уверенностью утверждать, что для разработки универсального интерактивного интерфейса взаимодействия между оператором и прикладной программой понадобятся все эти классы ввода. Поэтому, чтобы исключить потребность в специальной системе поддержки неграфического диалога, в состав функциональных возможностей GKS была включена возможность работать со всеми классами ввода.



Основные концепции (GKS). Концепция Ввода. Классы ввода.

Режимы работы ввода

Режимы ввода

Каждое логическое устройство ввода может функционировать в одном из трех возможных режимов. Режим работы определяется прикладной программой. В некоторый момент времени для обеспечения ввода с логического устройства можно использовать только один режим работы, например **ЗАПРОС, ОПРОС и СОБЫТИЕ**. В зависимости от режима значения вводятся оператором и передаются прикладной программе различными способами.

Режимы работы ввода

ЗАПРОС (REQUEST) — обращение к специальной функции в режиме ЗАПРОС вызывает попытку прочесть значение логического ввода с указанного логического устройства. GKS ожидает ввода до тех пор, пока оператор либо не введет данные, либо не выдаст сигнал отмены ввода, который зависит от устройства логического ввода и от реализации.

ОПРОС (SAMPLE) обращение к специальной функции в режиме ОПРОС предписывает GKS возвратить текущее значение логического ввода с указанного логического устройства, не дожидаясь действий со стороны оператора. Устройство должно находиться в режиме ОПРОС.

СОБЫТИЕ (EVENT) GKS поддерживает одну входную очередь событий, состоящую из упорядоченных в порядке поступления записей о событиях. Запись о событии содержит идентификатор логического устройства и значение логического ввода, полученное от этого устройства. Записи о событиях от устройств, работающих в режиме СОБЫТИЕ, генерируются асинхронно, исключительно в зависимости от действий оператора. Прикладная программа может получить из очереди запись о наиболее раннем событии и проанализировать ее содержание. Возможно удаление из очереди всех записей о событиях, касающихся определенного логического устройства ввода.

Основные концепции (GKS). Концепция Ввода. Классы ввода.

Режимы работы ввода

В режиме **ЗАПРОС** чтение данных с графической станции весьма похоже на чтение текстовой информации с терминала с помощью обычного оператора READ языка ВУ. В любой момент прикладной программой может быть запрошен ввод только с одного устройства. Это ведет к диалогу, полностью управляемому прикладной программой, т. е. находясь в этом режиме, оператор не может, например, в произвольный момент свободно ввести позицию или выбрать альтернативу. Оператор может прервать ввод с помощью оговоренного при реализации действия отмены. Например, таким действием может стать нажатие соответствующей клавиши на станции. В этом случае прикладная программа будет уведомлена, что произошло прерывание и ввод данных невозможен.

В режимах **ОПРОС** и **СОБЫТИЕ** оператор может управлять любым из нескольких устройств по собственному усмотрению. Все логические устройства ввода, находящиеся в этих режимах, доступны ему. Отличие заключается в том, что в режиме **ОПРОС** текущие значения опрашиваемых устройств известны прикладной программе независимо от того, изменял оператор данные или нет (или даже просто прикасался к устройству) , в то время как в режиме **СОБЫТИЕ** требуются определенные действия с его стороны, чтобы поместить значения в очередь событий.

Рассмотрим, например, изменение значений устройства ввода позиции с помощью джойстика в режиме **ОПРОС**. Данные, получаемые в результате опроса внутри цикла, используются для формирования новой матрицы преобразования и преобразования с ее помощью группы сегментов. Когда цикл выполняется достаточно быстро, у оператора создается впечатление, что изменение сегментов непосредственно связано с движением джойстика. Если при этом установить режим работы обычной и функциональной клавиатуры **СОБЫТИЕ**, то с помощью данных устройств он сможет оповестить программу о своем намерении завершить или продолжить ее выполнение. С этой целью прикладная программа в конце каждой итерации должна проверять очередь событий.

Основные концепции (GKS). Концепция Ввода. Классы ввода. Режимы работы ввода

Эхо, подсказки и инициализация устройств ввода

Когда прикладной программе нужно, чтобы оператор ввел данные, ей необходимо сообщить, что соответствующее действие требуется немедленно (в режиме **ЗАПРОС**) или ожидается (в режимах **ОПРОС** или **СОБЫТИЕ**). Такое приглашение оператора к некоторой активности называется **подсказкой**. Например, подсказкой может быть вывод на носитель изображения графических примитивов, текст "Введите значение" или результат действия функции сообщения. Однако существуют конкретные виды подсказок, привязанные к устройствам графического ввода. Прикладная программа может управлять этими видами подсказок. Для устройства ввода позиции подсказкой служит появление перекрестья или курсора, для устройства ввода текста появление курсора, а для устройства выбора, реализованного с помощью функциональной клавиатуры, - мигание светового сигнала.

Подсказка (Prompt) - выводимая оператору информация, указывающая на доступность данного логического устройства ввода.

В процессе взаимодействия **GKS** с устройством ввода последнее содержит значение, которое может изменить оператор. Текущее значение должно быть известно оператору. Такое оповещение называется **эхом**. Например, для устройств ввода позиции или последовательности позиций эхом служат следящее перекрестье или курсор, помещенные в текущую позицию, для устройства ввода строки визуальное представление на экране литер текста, для устройства указания мерцание последнего помеченного сегмента.

Эхо ВВОДА ПОЗИЦИИ

Следящее перекрестье



Курсор

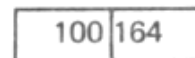


"Резиновая нить"

Текущая позиция



Числовое представление



Эхо ВВОДА ЧИСЛА

Указатель



Шкала



Числовое представление



Эхо (Echo) - немедленное оповещение оператора графического терминала о текущих значениях, которые обрабатываются устройством ввода. Примеры **Эха** (для у-в ввода позиции и числа):



Основные концепции (GKS). Концепция Ввода. Классы ввода.

Режимы работы ввода

Эхо, подсказки и инициализация устройств ввода

В режиме **ЗАПРОС** текущее значение передается прикладной программе только по явному указанию оператора, например, нажатием кнопки после установки нужного значения устройства ввода позиции. Такое же особое действие требуется для постановки в очередь текущего значения в режиме **СОБЫТИЕ**. Вид эха, информирующего оператора о том, что особое действие воспринято, называется ***подтверждением***.

Подтверждение (Acknowledgement) - вывод, оповещающий оператора логического устройства ввода, что его действие, отметившее определенный момент времени в процессе взаимодействия, принято системой во внимание.

Эхо и подтверждение, так же как и подсказку, реализуют логические устройства ввода, отображенные на физические устройства ввода. Управление всеми этими видами оповещения доступно прикладной программе, так как для определенного устройства ввода она может включить, или выключить эхо и подсказку, выбрать один из поддерживаемых устройством видов **подсказки/эха**, однако своих видов **подсказки/эха** она создавать не может. Эхо, заключающееся в изображении примитивов вывода прикладной программой в результате интерпретации его значения ввода, называется ***откликом*** (обратной связью). В предыдущем примере, когда преобразования сегмента определялись значениями, поступающими с устройства ввода позиции, изменения на экране размеров, ориентации и местоположения сегментов служили обратной связью перемещений позиционирующего устройства.

Отклик (Feedback) - вывод, оповещающий оператора о том, каким образом прикладная программа интерпретировала значение логического ввода.

Когда разрешается взаимодействие с устройством (т. е. когда запрашивается ввод или разрешается работа в режиме **ОПРОС** или **СОБЫТИЕ**), значение логического ввода инициализируется начальным значением, которое может быть задано прикладной программой. Режим **подсказки/эха**, если он включен, будет отражать это значение до тех пор, пока его не изменит оператор.

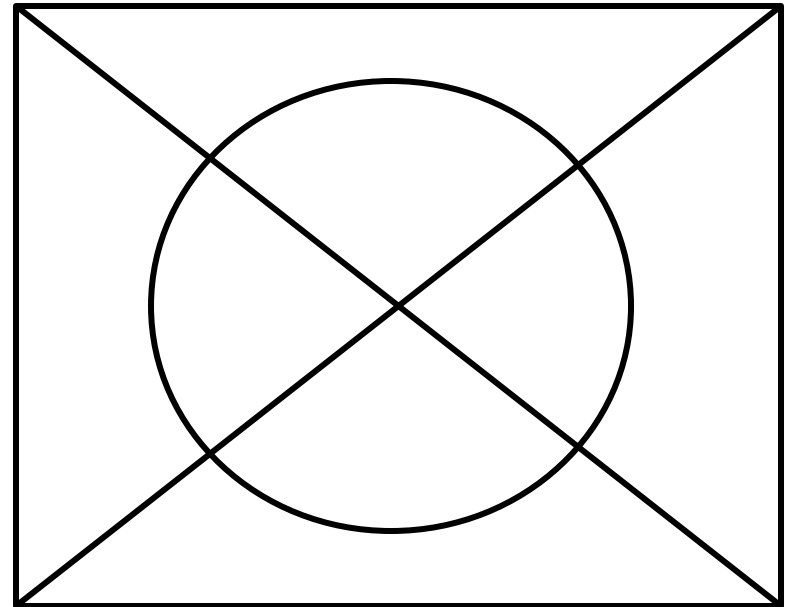
Основные концепции (GKS). Концепция Сегментации.



Важнейшая концепция МГ, определяющая возможность структурирования создаваемых изображений:

Изображение формируется из примитивов вывода. Они могут быть объединены в части, к которым можно обращаться и которыми можно манипулировать как единым целым. Эти части изображения называются **сегментами**.

Сегмент - это совокупность примитивов вывода, которой можно манипулировать как единым целым. Каждый **сегмент** сохраняется в памяти всех активных к моменту его создания графических станций. "Сохраняется" означает, что **сегмент** посылается на станцию, и далее его судьба зависит от преобразований сегмента, затребованных прикладной программой. Если графическая станция не обладает собственными средствами хранения, то GKS должна так управлять **сегментами**, чтобы создавалось впечатление, что память для них реально присутствует на станции.



Основные концепции (GKS). Концепция Сегментации.



Сегменты различаются по уникальному имени, называемому **именем сегмента**.

Все примитивы вывода объединяются в сегмент, начиная с момента его открытия и вплоть до его закрытия. После того как сегмент закрыт, ни один примитив не может быть добавлен к нему или удален из него. Создание нового сегмента невозможно до тех пор, пока не будет закрыт предыдущий.

Примитивы вывода внутри сегмента могут идентифицироваться дополнительным, не обязательно уникальным именем, которое называется **идентификатором указания**, являющимся частью значения ввода, возвращаемого (в прикладную программу) устройством указания после отметки сегмента.

Имя сегмента задается прикладной программой при его создании. Идентификаторы указания примитивов внутри сегмента задаются как статически устанавливаемый атрибут вывода.



Действия над сегментами

Основные концепции (GKS). Концепция Сегментации.



Действия над сегментами

Сегментами как единым целым можно манипулировать следующим образом:

1. Изменять их преобразование

Преобразованиями сегмента называют пересчет координат внутри пространства нормированных координат, заключающийся в переносе, масштабировании и повороте.

Преобразование сегмента преобразование, результатом которого является изменение на носителе изображения положения (перенос), размера (масштабирование) и/или ориентации (поворот) элементов изображения, определяемых сегментом.

Сдвиг – добавление постоянного смещения к положению всего или части изображения.

Масштабирование - увеличение или уменьшение размеров всего или части изображения путем умножения координат элементов изображения на постоянное значение. Для отдельного масштабирования по двум ортогональным направлениям требуется два постоянных коэффициента.

Поворот - поворот вокруг оси всего или части изображения.

Преобразования сегмента задаются связанной с ним матрицей преобразования. Последняя является матрицей размерности 2×3 , часть матрицы размерности 2×2 описывает масштабирование и поворот, а часть 2×1 - перенос. Для задания матриц преобразования прикладной программе доступны вспомогательные функции. Первоначально преобразование устанавливается единичным. Так как матрица преобразования сохраняется как часть описания сегмента, то начальное положение сегмента может быть восстановлено заменой матрицы преобразования на единичную.

Основные концепции (GKS). Концепция Сегментации.



Действия над сегментами

Сегментами как единым целым можно манипулировать следующим образом:

2. Изменять приоритет, выключать или включать видимость, чувствительность и выделение

3. Посылать копии на разные графические станции или вставлять их в другие сегменты

4. Удалять

5. Переименовывать

Сегмент может быть удален как со всех графических станций, на которых он хранится, так и только с одной указанной. После полного удаления сегмент неизвестен ни GKS, ни графическим станциям и его имя можно использовать повторно. Переименовывать сегмент означает изменять его имя на новое, не занятое ни одним из существующих сегментов.

Основные концепции (GKS). Концепция Сегментации.



Атрибуты сегмента

Атрибуты сегмента представляют собой характеристики его состояния, действие которых распространяется на все примитивы сегмента в целом.

К атрибутам сегмента относятся: **видимость, выделение, чувствительность, приоритет сегмента и его преобразование**. У каждого сегмента задаются свои значения атрибутов, которые не могут меняться при его представлении на различных графических станциях. У любого сегмента, включая и открытый, могут быть изменены атрибуты.

Выделение (Highlighting) независимый от устройства способ подчеркивания значения определенного сегмента путем изменения атрибутов, от которых зависит его визуальное представление.

Чувствительность (Detectability) атрибут сегмента, определяющий возможность применения в нем функции ввода, для указания объекта.

Видимость (Visibility) атрибут сегмента, определяющий присутствие сегмента на носителе изображения графических станций. Невозможно указать на невидимый сегмент.

Приоритет сегмента (Segment priority) - атрибут сегмента, используемый для определения, какой из нескольких перекрывающихся сегментов имеет предпочтение для графического ввода и вывода.

Действие приоритета распространяется только на видимые сегменты. Части примитивов, которые перекрываются примитивами другого сегмента с более высоким приоритетом, могут оказаться невидимыми. Когда указываются примитивы перекрывающихся сегментов, то выбирается сегмент с наибольшим приоритетом. Важнейшее значение имеет правильное вычисление приоритетов сегментов для раскрашенных или покрытых узором областей на растровых устройствах. В этих устройствах изображение строится на основе матрицы пикселей, каждый из которых может принадлежать только одному сегменту.

Основные концепции (GKS). Концепция Графического метафайла.




Ядро графической системы предлагает функции для записи изображения во внешний файл и извлечения его из файла.

Файлы, используемые для хранения изображений, называются **графическими метафайлами**.

Рабочая группа TC97/SC5/WG2 "Машинная графика" Международной организации по стандартизации выделила подгруппу по метафайлам, которая определила их следующим образом :

Графический метафайл это механизм для обмена и хранения графических данных в форме, независимой как от устройств, так йот приложений. Метафайл, применяемый в GKS, называется GKS-метафайлом или GKSM.

GKS-метафайл последовательный файл, который может быть считан или записан из GKS; используется для долговременного хранения (наряду с обменом и передачей) графической информации.



Среди наиболее важных причин, обусловивших введение графических метафайлов вместе с графической системой, можно назвать следующие:

Основные концепции (GKS). Концепция Графического метафайла.



- ✓ Должна иметься возможность воспроизводить графические данные на различных устройствах изображения. Необходимо предоставить пользователю выбор для представления его изображений на разных графопостроителях, устройствах вывода на микрофильм и экранах дисплеев.
- ✓ Графические данные должны быть восстанавливаемыми для последующего их использования. Они должны храниться независимым от устройств образом, чтобы устройство вывода изображения можно было выбрать после генерации данных.
- ✓ Графические данные должны быть переносимы как путем передачи их по линиям связи, так и обмена на носителях информации, например магнитных лентах. На большинстве вычислительных установок имеется несколько источников получения графических данных. Изображения, полученные в результате использования различных графических пакетов обработки изображений, численных моделей или экспериментальных данных, должны быть объединены единообразным представлением графических данных.
- ✓ И наконец, должен существовать какой-то способ редактирования уже созданной и сохраненной информации. Под редактированием понимается: изменение, удаление или добавление частей изображения, изменение их визуальных характеристик и слияние.
- ✓ Дополнительным важным преимуществом графических метафайлов является то, что они способны связывать различные графические устройства и системы стандартным и простым образом и тем самым позволяют осуществлять экономически эффективное коллективное использование дорогостоящего графического оборудования.



Формат GKSM

- **GKSM описывает плоские изображения. Каждое изображение представляется последовательностью записей данных (items), которые являются результатом вызова функций GKS. В метафайле содержатся записи для примитивов вывода и атрибутов вместе с заголовками файла, заголовками изображений и записью конца файла. Дополнительно к графическим записям GKSM может содержать и неграфические, зависящие от приложения данные.**
- **Метафайл формируется как последовательность логических записей переменной длины. Каждая запись начинается с ключа, определяющего ее тип. За ним следует значение длины данных в записи, так что она может быть легко пропущена интерпретатором, если соответствующий тип записи не представляет интереса. Формат данных для чисел в GKSM может выбираться в широких пределах.**
- **Представление во внутреннем машинном коде возможно так же, как и символьное представление в коде ASCII.**

Основные концепции (GKS). Концепция состояний GKS. Таблицы состояния.



➤ В любой момент выполнения программы GKS находится в некотором точно определенном состоянии. Состояние задается режимом функционирования и значениями переменных состояния, содержащимися в некотором числе имеющихся в GKS таблиц состояния. Действие функций частично определяется изменением значений в таблицах состояний, вызванным их выполнением.

➤ **Функциональное состояние может иметь** одно из следующих значений: «GKS закрыто»; «GKS открыто»; «есть открытая станция»; «есть активная станция»; «сегмент открыт».

➤ Переход из одного состояния в другое вызывается использованием функций управления в прикладной программе, например функция ОТКРЫТЬ GKS изменит состояние с «GKS закрыто» на «GKS открыто», если в момент вызова функции состояние было «GKS закрыто». Возможен вызов конкретной функции GKS или нет, зависит от состояния функционирования.

Все значения, содержащиеся в любой из таблиц состояния или описания ЯГС, могут быть получены в прикладной программе. Функции, используемые для этой цели, называются **справочными**.

Основные концепции (GKS). Концепция Обработки ошибок



- ☐ Для каждой функции ЯГС вместе с ее описанием приводится и список возможных ошибок.
- ☐ В каждую реализацию ЯГС необходимо включать обработку ошибок, способную обнаружить эти ошибки и поместить в файл сообщений запись хотя бы об одной из них.
- ☐ Записываются идентификатор ошибки и идентификатор функции ЯГС, ее обнаружившей.
- ☐ Прикладная программа может заменить стандартную процедуру обработки ошибок ЯГС на собственную, осуществляющую специальные реакции на ошибки.
- ☐ Ядро графической системы передает пользовательской процедуре идентификаторы как ошибки, так и обнаружившей ее функции ЯГС.
- ☐ С помощью справочных функций пользовательская процедура обработки ошибок может опрашивать таблицы состояния ЯГС.