

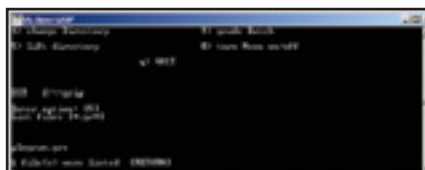


# РАЗРАБОТКА ПРИЛОЖЕНИЙ для **Unigraphics** НА ЯЗЫКЕ **GRIP**

## Построение исполняемого GRIP-приложения

Поговорим теперь о построении исполняемого модуля программы. Для этого снова запустим программную оболочку GRADE и при помощи четвертого пункта меню (**change directory**) изменим путь к папке, в которой сохранен исходный текст нашей программы PLOGRAM.GRS (в именах папок и файлов лучше избегать русскоязычных названий или имен, содержащих пробелы). Воспользовавшись пятым пунктом меню (**list directory**), убедимся, что программа существует в этой папке. Оболочка GRADE по умолчанию настроена таким образом, что выводит на экран список всех файлов, имеющих расширение **.gr\***. Таким образом в этот перечень попадают и исходные тексты с расширением **.grs**, и скомпилированные объектные модули с расширением **.gri**, и исполняемые программы, имеющие расширение **.grx**.

Предположим, вы сохранили исходный текст программы в файле с названием **plogram.grs** на диске D в папке с названием **grip**. В этом случае команда **list directory** выведет на экран список, состоящий из одного файла.



Возвращаемся к основному меню (клавиша **ENTER**) и приступаем к компиляции нашей программы. Среда разработки GRADE позволяет компилировать сразу несколько файлов с исходными текстами. Наш проект состоит из одного файла, после выбора второго пункта меню (**Compile**) указываем имя файла **plogram.grs** (расширение **.grs** в имени файла можно опустить).



Компилятор анализирует каждую строку программы и формирует достаточно обширный листинг, который содержит список всех переменных, меток и возможных ошибок (если вы набирали исходный код программы вручную, ошибки вполне вероятны). Строку с неверной записью компилятор метит в листинге символом **"\***". Показателем того, что программа скомпилирована без ошибок, будет сообщение **N GRIP PROGRAM COMPILED WITHOUT ERROR**. Как результат этого этапа в папке появляется файл со специальным объектным кодом, имеющий то же имя, что и файл с исходным текстом, и расширение **.gri**. При помо-

щи пятого пункта меню вы можете увидеть его в перечне файлов.

Третий, заключительный этап — построение исполняемого модуля из одного или нескольких объектных файлов. На сленге программистов этот процесс называется линкированием или линкованием (от английского link). В нашем примере основная программа не включает подпрограмм, весь исходный текст содержится в одном файле, а в результате компиляции сформирован один файл с объектным кодом. Для линкования выбираем третий пункт меню (**Link**) оболочки GRADE и указываем имя файла plogram.gri. Выводимый на экран листинг содержит перечень всех программных модулей, включаемых в исполняемое GRIP-приложение, сообщения об ошибках и другую полезную информацию. В случае успешного завершения процесса линковки (что подтверждается сообщением вида **N GRIP PROGRAM LINKED WITHOUT ERROR**) создается исполняемый файл, имеющий расширение .grx. Надеюсь, ваши труды увенчались успехом и исполняемый модуль с именем plogram.grx успешно сформирован.



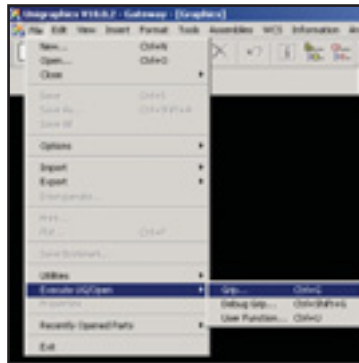
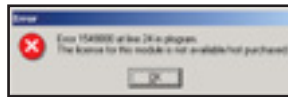
Формирование исполняемого модуля из нескольких объектных модулей имеет ряд особенностей, которые подробно изложены в разделе документации UG/Open.

### Запуск GRIP-приложения

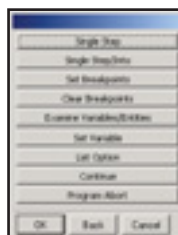
Как запустить созданную программу в сеансе Unigraphics? Запустите Unigraphics, создайте новый или откройте существующий Part-файл и выберите в меню *File* раздел *Execute UG/Open*; далее выберите либо запуск GRIP-программы в обычном режиме (пункт *Grip...*, ему соответствует комбинация горячих клавиш "Ctrl+G"), либо запуск в режиме отладки (пункт *Debug Grip...*, комбинация горячих клавиш "Ctrl+Shift+G"). При любом способе запуска вам будет предложено стандартное окно Windows для вы-

бора исполняемого файла с расширением .grx. Выбирайте программу plogram.grx — и стройте параллелограмм за параллелограммом!

Если вы не располагаете лицензией на выполнение GRIP-программ (Grip Execute), запуск GRIP-приложения невозможен. Программа выдаст примерно такое сообщение:



При создании сложных GRIP-приложений, состоящих из не-



скольких модулей, может оказаться полезным запуск приложения в режиме отладки — **Debug Grip**. В этом случае после выбора исполняемой программы пользователю будет предложено окно с командами отладки. Доступны пошаговый режим выполнения, получение информации о текущем состоянии переменных, включение точек приостановки программы (breakpoints), аварийное прерывание программы и другие полезные функции. Думаю, что программисты, которым потребуется режим отладки созданных ими GRIP-программ, без труда разберутся с его возможностями.

### Создаем панель инструментов для GRIP-программ

Надеюсь, созданная вами GRIP-программа успешно заработала. Утихли первые восторги... и им на смену пришел вопрос: "Неужели каждый раз для того чтобы постро-

ить параллелограмм придется проходить путь *File* → *Execute UG/Open* → *Grip* и разыскивать grx-файл в дебрях моего компьютера? За что бороться?!" Комбинация горячих клавиш "Ctrl+G" положение не спасает, поэтому предлагаю рассмотреть создание собственной панели инструментов, одна из кнопок которой будет активизировать наше GRIP-приложение для построения параллелограмма. Создаваемая панель инструментов напоминает стандартные панели Unigraphics (см. рисунок) и может включать любое количество кнопок, каждая из которых активизирует пункт меню Unigraphics, программу UG/Open, GRIP-программу или макрос.



Любая панель инструментов определяется в Unigraphics как внешний текстовый файл с расширением (.tbr), который содержит раздел описания заголовка панели инструментов и собственно раздел описания кнопок панели.

Создадим в любом текстовом редакторе файл grip.tbr следующего содержания:

```
!=====
! Панель инструментов для запуска GRIP-
! программ
! Название файла: grip.tbr
!
```

TITLE GRIP-программы

VERSION 170

```
BUTTON UG_GRIP_PLOGRAM
LABEL Построение параллелограмма
BITMAP D:\grip\bitmaps\plogram.bmp
ACTION D:\grip\plogram.grx
```

SEPARATOR

```
BUTTON UG_FILE_RUN_GRIP
LABEL Запуск Grip программы
BITMAP D:\grip\bitmaps\grip.bmp
```

```
BUTTON UG_FILE_RUN_GRIP_DEBUG
LABEL Запуск Debug Grip
BITMAP D:\grip\bitmaps\debug_grip.bmp
```

```
!===== Конец файла =====
```

Панель инструментов, создаваемая таким файлом grip.tbr, имеет заголовок "GRIP-программы" и содержит две группы кнопок, разделенные сепаратором. В первой группе всего одна кнопка — в качестве выполняемого действия мы определим ей запуск созданной нами программы построения параллелограмма plogram.gfx. Во вторую группу входят две кнопки, одна из которых запускает выбранную пользователем GRIP-программу, а вторая производит те же действия в режиме отладки.

Каждая кнопка имеет уникальный идентификатор BUTTON, определение всплывающей подсказки LABEL (эта подсказка появляется в момент наведения указателя мыши на кнопку панели инструментов), определение BITMAP для изображения кнопки и описание выполняемого этой кнопкой действия ACTION.

Несколько подробнее об изображении BITMAP для кнопок меню. Казалось бы, всё просто: после определения BITMAP указан путь к bmp-файлу, который и будет выводиться в качестве изображения кнопки. Но Unigraphics позволяет отображать как стандартные панели инструментов, так и созданные пользователем (с кнопками размером 16x16, 24x24, 32x32 и 48x48 пикселей, причем как цветными, так и монохромными). Для каждой кнопки должны быть представлены четыре растровых файла: 16x16 монохромный, 16x16 цветной, 24x24 монохромный и 24x24 цветной. Изображения кнопок 32x32 и 48x48 формируются в системе Unigraphics автоматически.

По этой причине запись в определении BITMAP указывает путь к группе растровых bmp-файлов. В нашем примере определение plogram.bmp указывает на четыре файла:

plogram_lm.bmp	24x24, монохромный (lm – Large, Monochrome)
plogram_lc.bmp	24x24, цветной (lc – Large, Color)
plogram_sm.bmp	16x16, монохромный (sm – Small, Monochrome)
plogram_sc.bmp	16x16, цветной (sc – Small, Color)

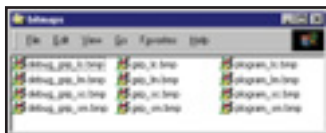
Создать bmp-файлы можно при помощи растрового редактора Paint из стандартного набора инструментов Windows или любого другого (Adobe PhotoShop, CorelDraw и т.п.). На иллюстрации приведен пример изображения цветной кнопки 24x24 plogram\_lc.bmp с использованием редактора Paint. Не судите



строго мои художественные способности, по замыслу эта фигура должна изображать параллелограмм. Пространство 24x24 пикселя (а тем более 16x16) несколько сдерживает полет фантазии, и хорошо нарисованную пиктограмму можно рассматривать как произведение художественной миниатюры!

Достаточно нарисовать цветные изображения кнопок 16x16 и 24x24, а затем средствами графического редактора создать их монохромные копии.

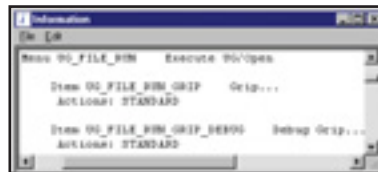
Итак, для трех кнопок меню необходимо создать 12 bmp-файлов,



имена которых формируются по описанным выше правилам. Как выглядит в результате содержимое папки bitmaps, показано на рисунке.



Несколько слов об уникальном для каждой кнопки идентификаторе BUTTON. Если для первой кнопки, которая запускает наше собственное приложение, его можно определить абсолютно произвольно, то для кнопок, которые дублируют стандартные пункты меню (File → Execute UG/Open → Grip... или Debug Grip...), идентификатор BUTTON должен соответствовать вполне определенному значению. Как узнать это значение? Для этого в меню Unigraphics выберите пункт Information → Custom Menubar → Report Tool..., укажите, что желаете получить полную информацию (пункт Complete Menubar) по основному меню Unigraphics (UG\_GATEWAY\_MAIN\_MENUBAR) и нажмите кнопку OK. Полный список всех разделов меню с идентификаторами BUTTON (Item) будет выведен в информационное окно. Список достаточно длинен, и, чтобы найти требуемый пункт меню, придется воспользоваться средствами поиска. В нашем примере интерес представляют строки с пунктами меню "Grip..." и "Debug Grip..."



Выбираем из полученного списка нужные пункты меню (Item) — и их идентификаторы (например, UG\_FILE\_RUN\_GRIP) вставляем как соответствующие идентификаторы BUTTON в наш файл панели инструментов grip.tbr. Обратите внимание, что при создании кнопки панели инструментов, которая дублирует пункт основного меню и вызывает стандартную процедуру Unigraphics, выполняемое действие ACTION для такой кнопки не указывается.

В окончательном варианте панель инструментов для запуска GRIP-программ будет выглядеть следующим образом:



Файл нашей собственной панели инструментов сформирован. Конеч-

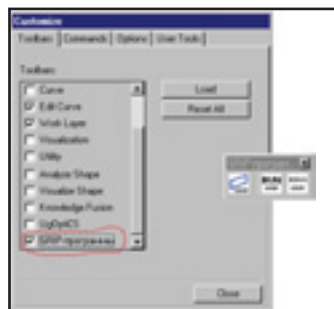
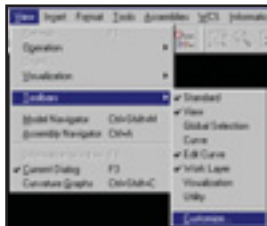


но, вы можете добавить в него и другие кнопки, создать вложенные многоуровневые меню (о том, как это сделать, — читайте в документации UG/CAD раздел "Gateway/Toolbars") и назначить им соответствующее действие.

### Загрузка панели инструментов

Загрузка и активизация созданных пользователем панелей инструментов может производиться двумя способами.

1. В меню Unigraphics выберите пункт *View → Toolbars → Customize...* (см. рисунок). В предложенном диалоговом окне нажмите кнопку *Load*, выберите загружаемый файл панели инструментов (*grip.tbr*). Список всех имеющихся в системе активных и скрытых панелей инструментов пополнится названием новой подгружаемой панели (в нашем примере — "GRIP-программы"). При первой загрузке панель автоматически появится на экране, после чего (в пределах сеанса работы) она скрывается или отображается вновь при помощи флажка, расположенного слева от названия панели. Размещение панели инструментов в пределах окна Unigraphics подчиняется законам Windows, положение панели запоминается в системе, однако при следующем запуске Unigraphics процесс загрузки созданной пользователем панели инструментов придется повторить.



2. Для того чтобы пользовательская панель инструментов автоматически загружалась при каждом старте Unigraphics в файл *custom\_dirs.dat*, находящийся в каталоге *\$UGII\_BASE\_DIR\$\ugii\menus*, включите путь к каталогу, в котором располагается папка *startup*, где в свою очередь сохранен файл с пользовательской панелью инструментов (*grip.tbr*). Обратите внимание: *необходимо указать путь к папке startup*, а не к файлу *grip.tbr*. Соответственно, если в папке *startup* находится несколько файлов (*.tbr*), все панели инструментов, определенные в этих файлах, будут загружены. Ниже приведен фрагмент файла *custom\_dirs.dat*:

```
# Customers should feel free to edit this file.
#
#####
#
# Customer modifications can follow on here
#
D:\grip
##### Конец файла #####
```

Строго говоря, создание собственных панелей инструментов не относится к теме программирования на языке GRIP и вообще не входит в раздел UG/Open, но для рассказа о правилах построения пользовательских панелей инструментов этот раздел подходит как нельзя лучше. Позже мы добавим в эту панель кнопку для запуска приложения User Function.

Если вам удалось повторить примеры, рассмотренные выше, или заработала ваша первая самостоятельно разработанная GRIP-программа, — знакомство с основами этого простого, но достаточно гибкого и мощного языка для создания собственных графических приложений можно считать состоявшимся.

Юрий Чугушев  
Consistent Software  
Тел.: (095) 913-2222  
E-mail: jura@csoft.ru

## НОВОСТИ

### Новая версия системы Unigraphics выходит на российский рынок



На российский рынок вышла новая версия системы Unigraphics, получившая название NX2. Программный продукт поставляется полностью локализованным, с переведенной на русский язык документации.

Unigraphics — высокоуровневая комплексная система, охватывающая все этапы создания изделия: от концептуального дизайна и проектирования, выполнения инженерного анализа до подготовки производства. Направление ее развития компания-разработчик обозначила как "стратегию NX": Unigraphics всё более полно решает комплекс задач проектирования и изготовления новых изделий.

"NX2 — следующий шаг в совершенствовании наших систем. Ряд важных улучшений позволил сделать намного более удобными инструменты проектирования, подготовки производства и инженерного анализа. Появились дополнительные возможности в использовании языка инженерных знаний, расширена интеграция с системой поддержания жизненного цикла изделия Teamcenter", — сказал Чак Гриндстафф (Chuck Grindstaff), руководитель подразделения PLM-продуктов компании EDS PLM Solutions.

Улучшения, внесенные в систему, призваны сократить время выбора верного решения при проектировании с использованием цифровых технологий. Помимо традиционных выгод от применения систем CAD/CAM/CAE (уменьшение стоимости изделия, сокращение сроков его выхода на рынок), NX позволяет производителям сосредоточиться на многократном использовании накопленных знаний, что улучшает качество изделия и позволяет применять инновационные технологии.

Поставки NX2 российским пользователям начались в третьем квартале этого года.