

федре психологии и безопасности жизнедеятельности ТРТУ. В настоящее время в рамках лабораторных и индивидуальных занятий со студентами, изучающими курс БЖД, используется АОКС "Производственный травматизм", которая позволяет изучать такие законодательные документы как "Положение о порядке расследования и учета несчастных случаев на производстве", "Кодекс законов о труде Российской Федерации" и получать навыки их использования при решении ситуационных задач. Аналогичная система АОКС разработана и внедрена по разделу курса "Обеспечение безопасности в чрезвычайных ситуациях". Она предназначена для освоения "Норм радиационной безопасности" и "Основных санитарных правил ОСП-72/87", а также для изучения поведения человека в нештатных ситуациях, принятия им правильных решений.

Значительно повышает информационное обеспечение и дает возможность конкретизировать применяемые мероприятия по биологически значимым факторам производственной среды и трудового процесса новая "Автоматизированная система анализа и учета условий труда".

Данная система позволяет: изучать основные нормативные документы по терминологии в области безопасности труда и регламентации опасных и вредных производственных факторов (СН, СНиП, ГОСТы ССБТ и др.); производить аттестацию условий труда на рабочем месте; получать оперативные данные о состоянии условий труда на предприятии в целом и на любом из рабочих мест, а также осуществлять прогноз заболеваемости персонала; делать выводы о приоритетности проведения профилактических и оздоровительных мероприятий как на предприятии в целом, так и на отдельно взятом рабочем месте.

Данная система вырабатывает у студента взгляд на необходимость решения проблем безопасности труда в производственных условиях, обеспечивает его нормативно-законодательной базой в виде эталонного банка данных, учит принятию обоснованных и эффективных мероприятий по улучшению условий труда. Кроме того, разработанная автоматизированная система может использоваться и для практических целей на производстве в системе автоматизированного рабочего места специалиста по охране труда.

УДК 681.3.069

С.И. Лебедь

ТЕХНОЛОГИЯ СОЗДАНИЯ ДИНАМИЧЕСКИХ ДЕМОНСТРАЦИОННЫХ РОЛИКОВ ПРИКЛАДНЫХ ПРОГРАММ, НАПИСАННЫХ НА ЯЗЫКЕ TURBO PASCAL 7.0 С ПРИМЕНЕНИЕМ БИБЛИОТЕКИ КЛАССОВ TURBO VISION

В настоящее время одной из самых актуальных проблем, встающих при разработке прикладных программ, является создание средств обучения пользователей работе с ними. К наиболее полезным из таких средств относятся демонстрационные ролики. К сожалению, решение проблемы создания демонстрационного ролика программного продукта зависит от большого числа обстоятельств, в частности, языка программирования, и не может быть универсальным. В данной статье рассматривается вопрос о создании динамических демо-роликов программ, написанных на языке

программирования Turbo Pascal версии 7.0 с использованием библиотеки классов Turbo Vision. Для решения этой задачи был написан TPU-модуль POWERAPP.TPU, первая версия которого позволяет с малыми затратами времени (около 10 мин) добавить программе способность (дополнительно к основным функциям) генерировать демонстрационные ролики, а также воспроизводить их.

1. О принципах технологии использования модуля POWERAPP

Предлагаемая вашему вниманию технология создания демонстрационных роликов основана на принципе централизованного сбора событий в программах, написанных на основе библиотеки Turbo Vision. В контексте данной библиотеки под событием понимают что-либо, на что программа должна отреагировать. Например, нажатие клавиши на клавиатуре или перемещение манипулятора "мышь". При использовании модуля POWERAPP, в режиме записи демо-ролика прикладная программа записывает информацию обо всех поступающих от клавиатуры или от "мышки" событиях в файл, отслеживая временные задержки между событиями. В режиме воспроизведения ролика, программа читает из файла информацию о событиях и задержках между ними и имитирует своевременное наступление соответствующих событий. Поэтому динамика изменения информации на экране при воспроизведении ролика будет такой же, как и при записи.

2. Ограничения, накладываемые на использование модуля

Программа, в которой программист собирается использовать модуль POWERAPP.TPU, должна отвечать требованиям, изложенным ниже:

1. Центральным объектом программы должен являться объект, порожденный от типа TApplication. Будем в дальнейшем называть этот объект **приложением**.

2. Если программист перекрывал метод GetEvent предка приложения, то этот метод должен начинаться следующим вызовом:

```
inherited GetEvent(Event),
```

как это обычно и бывает.

3. На протяжении всего времени выполнения программы прием информации с внешних устройств должен осуществляться централизованно в методе GetEvent приложения, причем множество внешних устройств ввода должно быть ограничено клавиатурой и манипулятором "мышь".

Этим требованиям отвечает подавляющее большинство программ, написанных на Turbo Vision. Кроме того, они еще менее критичны, чем это может показаться, так как на самом деле они относятся не ко всей программе в целом, а только к той последовательности команд (ветви программы), которая будет выполняться в процессе записи демонстрационного ролика. Например, если в вашей программе реализован так называемый DOS Shell (временный выход в ДОС), то это не значит, что вы не можете использовать модуль POWERAPP, а всего лишь значит, что вы не сможете проиллюстрировать выполнение DOS Shell в вашем демонстрационном ролике. Но без этого часто можно и обойтись.

3. Создание демонстрационного ролика

Для создания демонстрационного ролика необходимо:

- 1) тщательно продумать подробный сценарий демонстрационного ролика.

2) запустить прикладную программу в режиме генератора демо-ролика. Для этого необходимо в качестве **последнего** параметра командной строки указать ключевой параметр:

```
/MakeDemo:[path]demofile ,
```

где [path]demofile — спецификация создаваемого файла демо-ролика;

/ — любой символ (на ваш вкус)

3) действуя по сценарию, записать демонстрационный ролик.

При работе с программой в режиме записи ролика необходимо учитывать, что все задержки между нажатиями клавиш будут отражены один к одному в режиме воспроизведения демо-ролика. Поэтому, если возникла вынужденная задержка с вводом информации (например, перекур), необходимо нажать **клавишу паузы при записи** (по умолчанию это Alt-Space). Выход из режима паузы — любая клавиша. Время паузы будет "вырезано" из демо-ролика.

4. Воспроизведение демонстрационного ролика

Чтобы воспроизвести демонстрационный ролик, необходимо запустить прикладную программу с теми же параметрами, что и при записи демо-ролика, заменив лишь **последний** (т. е. /MakeDemo) на симметричный:

```
/PlayDemo:[path]demofile ,
```

где [path]demofile — спецификация создаваемого файла демо-ролика,

/ — любой символ

Воспроизведение демонстрационного ролика является управляемым.

По умолчанию определены следующие клавиши управления:

Space — пауза. Продолжение — любая клавиша.

[@] — пошаговое ускорение выполнения ролика

[¬] — пошаговое замедление выполнения ролика

ESC — прервать выполнение ролика.

5. Преобразование прикладной программы

Для преобразования программы в генератор и проигрыватель демонстрационных роликов необходимо выполнить следующие шаги.

1) Добавить модуль POWERAPP в список USES прикладной программы.

2) Везде в тексте программы заменить идентификатор типа Tapplication на TPowerApplication, используя, например, возможность "поиск с заменой" любого текстового редактора. Скорее всего, такая замена будет только одна.

3) В демонстрационном режиме к строке вызова программы добавляется дополнительный параметр, о котором самой программе лучше не догадываться. Поэтому при использовании в программе стандартной функции ParamCount необходимо заменить все ее вызовы на вызовы одноименного метода вашего приложения, определенного как TPowerApplication.ParamCount.

Однако при вызове ParamCount внутри какого-либо метода вашего приложения нет необходимости заменять данный вызов: Pascal и без этого правильно отработает данную ситуацию.

6. Обработка ошибочных ситуаций

При использовании демонстрационных возможностей программы могут возникать два типа ошибок при записи демо-ролика и два типа ошибок при его выполнении. Для них в модуле POWERAPP определены следующие константы:

```
dhmOpenDemoError = 1;    {ошибка открытия файла демо-ролика}
dhmCreateDemoError = 2;  {ошибка создания файла демо-ролика}
dhmReadDemoError = 3;    {ошибка чтения файла демо-ролика}
dhmWriteDemoError = 4;   {ошибка записи файла демо-ролика}
```

При возникновении ошибочной ситуации приложение автоматически вызывает виртуальный метод DemoHaltMsg, указав в качестве его параметра одну из приведенных констант, после чего аварийно завершает программу. При этом возвращается нулевой код ошибки, чтобы не было путаницы с кодами ошибки, используемыми вами.

Метод DemoHaltMsg определен для типа TPowerApplication следующим образом:

```
procedure DemoHaltMsg(Error: Integer); virtual;
```

Этот метод по умолчанию в зависимости от кода ошибки выдает пользователю соответствующее сообщение и при необходимости может быть перекрыт программистом.

УДК 574:378.1

Н.А. Лызь

О КОНЦЕПТУАЛЬНЫХ ОСНОВАХ СОДЕРЖАНИЯ ЭКОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ

Человеческое общество на современном этапе своего развития столкнулось с глобальными экологическими проблемами, которые по своей значимости для судеб человеческой цивилизации вышли на первое место. Темпы деградации природного окружения дают более чем достаточные основания для тревоги за будущее человечества и всей планеты. Требованием современности стало всеобщее экологическое образование во всех формах воздействия на сознание и поведение людей. Здоровая жизненная среда должна быть заботой всех.

Необходимо отметить, что в системе высшего образования в России отсутствует концептуально и содержательно выстроенная система экологического образования, отвечающая современным требованиям. Существуют лишь отдельные дисциплины в учебных планах высших учебных заведений (как правило, естественно-научного цикла), в которых излагается проблематика охраны окружающей человека среды. Учитывая комплексность экологических проблем, которые одновременно могут затрагивать области биологии, географии, физики, химии, медицины, политики, экономики, техники, философии, связывая воедино знания естественных, общественных и технических наук, представляется целесообразным рассматривать экологическое образование как особый срез, "сквозную" от-