

СРЕДСТВА ВИЗУАЛИЗАЦИИ ПРОЦЕССА НАПИСАНИЯ ПРОГРАММНОГО КОДА ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

TOOLS FOR VISUALIZATION THE PROCESS OF WRITING PROGRAM CODE IN PROGRAMMING TEACHING

Юри Тодоров Хоптериев Yuri Todorov Hopteriev

ассистент

yurih@uni-plovdiv.bg

Пловдивский университет имени Паисия
Хилendarского, Пловдив, Болгария

University of Plovdiv "Paisii
Hilendarski", Plovdiv, Bulgaria

Аннотация. Рассматривается Windows-приложение, предназначенное для динамического представления примеров в обучении программированию. Приложение предоставляет преподавателю возможность подготовить и сохранить решения примерных задач (программы/скрипты) в комплекте со специальными данными, необходимыми для их динамического представления. С помощью мультимедийного проектора преподаватель может неоднократно демонстрировать студентам процесс решения заранее подготовленной примерной задачи.

Ключевые слова: программирование, обучение, преподавание, динамическое представление примеров, мультимедиа.

Abstract. This paper describes a Windows application designed for a dynamic presentation of examples in programming teaching. The application gives the teacher an opportunity to prepare and save the solutions of example problems (programs/scripts), complete with special data necessary for their dynamic presentation – in writing and testing process. By means of a multimedia projector the teacher can repeatedly demonstrate to students the solution process of a previously prepared example problem.

Keywords: programming, learning, teaching, dynamic presentation of examples, multimedia.

«В современном мире цифровые технологии проникают во все сферы жизни, в том числе и в систему образования» [1, с. 113]. В обучении используются разные программные и технические средства. Одним из предпочтительных технических средств является мультимедийный проектор. Многие преподаватели заменяют написание на доске заранее подготовленными слайдами, которые можно многократно визуализировать студентам с помощью мультимедийной системы. В учебном процессе мультимедиа предоставляет дополнительные возможности

для лучшего представления учебного материала. Одной из таких дисциплин является программирование. Как известно, практические задачи в обучении программированию реализуются на компьютере, а компьютер является частью мультимедийной системы, что позволяет преподавателю, использующему мультимедиа, не только показать студентам готовое решение примерной задачи (программу/скрипт), но и продемонстрировать процесс ее решения: написание, тестирование и отладку. Но если демонстрация этого процесса выполняется пол-

ностью вручную, проявляются некоторые недостатки этого способа преподавания [2]:

- ручной ввод программного кода преподавателем замедляет его работу;
- одновременное выполнение двух действий, а именно ввод программного кода и объяснение решения, требует высокой степени концентрации;
- при проведении одного и того же упражнения с несколькими группами студентов необходимо вводить один и тот же код, т. е. выполнять одну и ту же рутинную работу несколько раз.

Было бы хорошо свести к минимуму рутинную работу.

Процесс написания кода программы может быть записан как видеоролик, который можно показывать студентам, но это не то же самое. Программный код в видеоролике нельзя редактировать или дополнять во время занятия, и, что более важно, программу из видеоролика нельзя напрямую компилировать и выполнить, так как она не находится в текстовом виде. Было бы лучше, если бы программный код вводился автоматически в текстовом редакторе. Эта цель может быть достигнута путем создания специализированного программного обеспечения. В статье рассматривается Windows-приложение, разработанное автором и предназначенное для динамического представления (в процессе написания и тестирования) решений примерных задач (программ/скриптов) в обучении программированию.

Динамическое представление примеров

Идея автоматизированного динамического представления решений примерных задач (программ/скриптов) рассмотрена и реализована в нескольких разработках [2, 3, 4]. Код решения заранее подготовленного примера автоматически пишется по шагам и сериям шагов. Шаг – это посимвольное написание или удаление последовательности символов (в частности, одного символа), представляющей собой небольшую часть решения [3]. Серия – это последовательность шагов (в частности, один шаг), которая выполняется без перерыва во времени [3]. Каждый шаг можно вставить или удалить в уже написанном коде. Отслеживая пошаговое написание программы, студенты имеют возможность наблюдать, каким образом вводит код опытный

программист. Например, когда программист пишет блок операторов C++ в текстовом редакторе, он обычно сначала записывает открывающую и закрывающую фигурные скобки, а затем вставляет операторы между ними [4]. Порядок выполнения шагов устанавливается заранее (при подготовке примеров) автором учебного материала. Решения более сложных задач могут быть представлены поэтапно. Этап представляет собой последовательность серий (в частности, одна серия), которая может рассматриваться как законченная часть решения [3]. После окончания каждого этапа может отображаться результат.

Windows-приложение для динамического представления примеров

Основными инструментами рассматриваемого приложения являются демонстратор примеров (рис. 1) и редактор для подготовки данных (рис. 2).

Демонстратор примеров (см. рис. 1) предоставляет возможность преподавателю, используя мультимедийный проектор, представить студентам динамически (как описано выше) решения заранее подготовленных примерных задач. Демонстрация запускается и управляется соответствующим образом преподавателем, который объясняет логику решения. После завершения демонстрации редактор становится доступным для набора текста на клавиатуре.

В своей работе демонстратор примеров взаимодействует с текстовым редактором приложения, в котором пишется решение примерной задачи.

Основными компонентами демонстратора являются (см. рис. 1):

- кнопка «Load» для загрузки нового примера;
- кнопка «Write» для автоматического набора в редакторе очередной серии шагов;
- кнопка «Result», с помощью которой показывается результат выполнения решения (программы/скрипта) или его этапа.

Для работы кнопки «Result» надо заранее установить необходимое базовое программное обеспечение для соответствующего языка программирования и сделать необходимые настройки приложения. Например, для выполнения показанной программы на языке C++ (см. рис. 1) приложение использует GNU-компилятор.

В этом случае при нажатии на кнопку «Result» выполняются последовательно два действия: сначала запускается компиляция программы, а потом, при успешной компиляции, выполняется скомпилированная программа. То же самое можно сделать и посредством нажатия на элемент «Execute» главного меню приложения.

Редактор данных (см. рис. 2) предоставляет возможность преподавателю или автору учебного материала подготовить и сохранить данные, необходимые для динамического представления примеров.

Основными компонентами редактора данных являются (см. рис. 2):

- кнопка «New» для начала нового примера;
- кнопка «Set step» для установки шага;
- кнопка «End of series» для установки конца серии;
- кнопка «Save» для сохранения данных;
- кнопка «Load» для загрузки данных.

В своей работе редактор данных взаимодействует с текстовым редактором приложения, в котором вводится или загружается из файла код решения (программа/скрипт).

Содержание и порядок шагов устанавливаются автором учебного материала. Последовательно (в желаемом порядке) содержание каждого шага выделяется в коде решения с помощью мыши (см. рис. 2) и посредством кнопки «Set step» включается в список шагов. С помощью кнопки «End of series» устанавливается конец текущей серии шагов, который

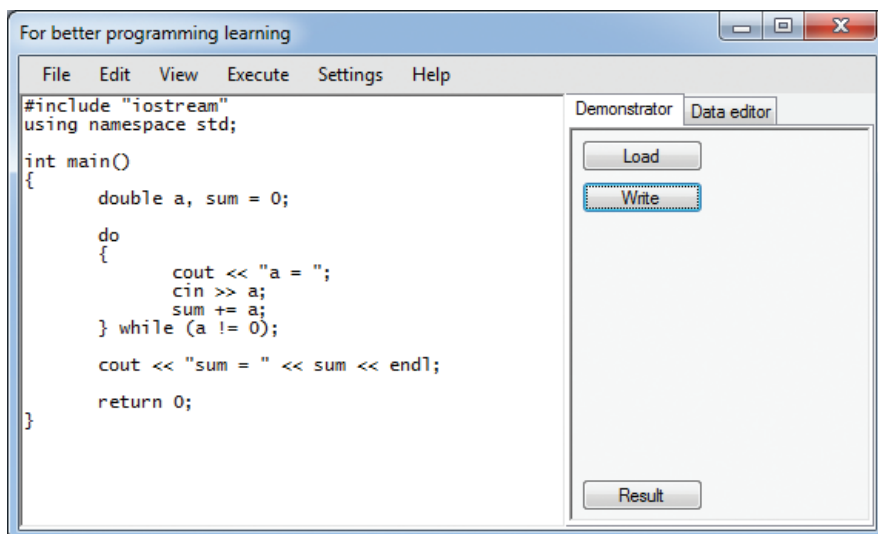


Рис. 1. Демонстратор примеров

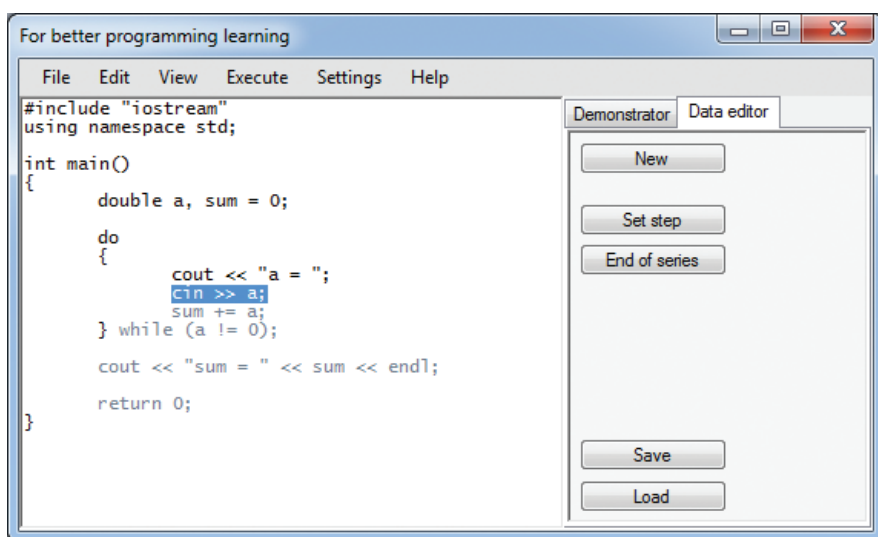


Рис. 2. Редактор данных

также является началом следующей серии. Данные о шагах и сериях вместе с кодом решения сохраняются в файле. Подготовленные примеры могут быть использованы как в Windows-приложении, так и в модулях, предназначенных для включения в электронные учебники и курсы в web-формате [2–4].

«Одним из очевидных достоинств мультимедийного учебного занятия является многократное усиление эффекта наглядности» [5, с. 347]. Рассмотренное приложение способствует именно наглядному представлению процесса написания программ и скриптов. Динамическое представление решения примерной задачи в сочетании с хорошим объяснением преподавателем дает студентам возможность понять детали решения, а также изучить некоторые лучшие практики, показанные преподавателем при представлении решения.

Приложение реализовано на Visual Basic и работает под Windows 7 и выше. Предстоит совершенствование редактора данных. Предусмотрена возможность автоматического генерирования части шагов по заранее заданным правилам, что значительно снизит рутинную работу по подготовке примеров.

Список литературы

1. *Дорожкин, Е. М.* Подготовка педагогов профессионального обучения в условиях цифрового образования: особенности, проблемы, пути решения / Е. М. Дорожкин, А. О. Прокубовская, Е. В. Чубаркова. Текст: непосредственный // Новые информационные технологии в образовании и науке. 2019. Вып. 2. С. 113–117.
2. *Хоптериев, Ю. Т.* Демонстрирование примеров в обучении веб-программированию / Ю. Т. Хоптериев. Текст: непосредственный // Известия Союза ученых – Сливен (Болгария). 2016. Т. 31 (1). С. 29–32 (на болгарском).
3. *Хоптериев, Ю. Т.* Модуль для динамического представления примеров в электронных учебниках по веб-программированию — способ реализации / Ю. Т. Хоптериев. Текст: непосредственный // Фундаментальные науки и применения: журнал Технического университета в Пловдиве. 2006. Т. 13 (1). С. 185–190 (на болгарском).
4. *Hopteriev, Y. T.* A dynamic presentation and online testing of examples in programming e-training courses / Y. T. Hopteriev. Text: print // Information technology. Problems and solutions. Ufa: USPTU. 2020. V. 4 (13). P. 34–40.
5. *Лукинов, В. А.* Аспекты визуализации эмпирического базиса учебной дисциплины / В. А. Лукинов, Д. А. Соколов. Текст: непосредственный // Наука. Информатизация. Технологии. Образование: материалы 13-й Международной научно-практической конференции «Новые информационные технологии в образовании и науке НИТО 2020». Екатеринбург, 2020. С. 346–356.