РАЗДЕЛ V.
ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В ОБУЧЕНИИ МАТЕМАТИКЕ В ШКОЛЕ И ВУЗЕ

И.А. Иванов, С.И. Иванова, М.Н. Иванова,
П.А. Корниенко, (Сочи), В.В. Орлов (С.-Петербург)

ПРОГРАММА *3DS-GEOMETRY* ПОСТРОЕНИЯ 3*DS*-ИЗОБРАЖЕНИЙ ГЕОМЕТРИЧЕСКИХ СТРУКТУР С ПРИМЕНЕНИЕМ ВХОДНОГО ЯЗЫКА *LSDSS* И ЕЕ ВОЗМОЖНОСТИ В ОБУЧЕНИИ СТЕРЕОМЕТРИИ В ШКОЛЬНОМ КУРСЕ МАТЕМАТИКИ

В интернет-издании www.cnews.ru [1] появилась информация о создании языков «для общения IT-шников с нормальными людьми». Речь идет о разработке институтом развития интернета (ИРИ) интернет-языка, “который позволит вести межотраслевые коммуникации и понятно общаться друг с другом”. Расширяя эту идею в контексте решения проблем школьного математического образования, хотелось бы получить «доступные *средства программирования* для пользователя», т.е. такие средства и интерфейс, которые максимально приближены к «естественному» математическому языку. Как известно, разработка интерфейсов, повышающих эффективность использования программных продуктов для решения профессиональных задач, – одна из основных задач, которая стоит перед разработчиками программного обеспечения. В настоящее время предложено достаточно большое количество программных продуктов, используемых в процессе обучения математике в школе, и, в частности, стереометрии. В этих продуктах реализован «классический» вариант представления геометрических конструкций – формат 3*D* (изображение объемных структур на плоскости). При этом «картинку» можно масштабировать, вращать, изменять ракурс для получения наилучшего представления об «объемной» фигуре. В нашем исследовании предлагается новый программный продукт 3*Ds-Geometry*, написанный на языке программирования *Python 3*.3. Этот продукт позволяет учителю и ученику строить 3*Ds*-изображения (*стереометрические*) геометрических конструкций, при этом используется «входной язык», приближенный к «естественному математическому». Таким образом, предлагается программный продукт, позволяющий строить *3D*s-изображения (просматриваемые через анаглифические очки) геометрических структур с использованием понятного для любого школьника, учителя, просто конечного пользователя *входного языка LSDSS* (*Language for the search and development of the stereoscopic structures* – язык для исследования и разработки стереоскопических структур), предназначенного для общения *user* с компьютером. Некоторые аспекты проблематики построения 3*Ds*-изображений были рассмотрены ранее, например, в [2].

При разработке входного языка использовался принцип *минимальности* – для построения стереоизображения применяются *простейшие геометрические объекты* – линии, точки; *основные арифметические операции* (+, –, *, /, ** –

сложение, вычитание, умножение, деление, возведение в степень) и *одна* функция – *sqrt*(*x*). Очевидно, что этот принцип ограничивает возможности программы, но в качестве достоинства может быть признана *простота* его изучения и применения входного языка для построения 3*Ds*-изображений достаточно сложных объемных геометрических структур.

1. Входной язык *LSDSS* и его синтаксис. Рассмотрим описание основных элементов входного языка – операторов, структур, выражений и переменных.

1.1. *Работа с операторами и структурами.* Входной язык *LSDSS* используется программой для построения изображений. Преимущество языка *LSDSS* перед языками программирования заключается в том, что он доступен и прост в использовании, количество элементов языка – минимально: 5 «операторов» и 1 структура.

Операторы: *line* A B (линия между точками A и B), *width* 5 (задание толщины линии, например, 5*px*), *color* red (цвет линии, по умолчанию *red*), *dash* / *solid* (*dash* для задания пунктирных линий, *solid* – сплошных).

Единственная структура, используемая языком – ***структура*** *point*, позволяющая задавать точки на плоскости позиционирования в компьютерной системе координат (*x*, *y*), ось *x* направлена стандартно (вправо), ось *y* направлена вертикально вниз. Для того, чтобы задать точку с именем *point_name* с координатами (*x*, *y*) во входном файле требуется написать point_ *name* (*x*, *y*). Например, точку *A* с координатами *x* = 100, *y* = 200 следует задавать следующим образом: *A*(100, 200). В данной конструкции обязательно должно быть указано имя точки (имя должно содержать хотя бы одну заглавную латинскую букву, в имени можно использовать цифры), должна присутствовать запятая, разделяющая координаты *x* и *y*, а также скобки, отделяющие имя точки от ее координат. Пробелы игнорируются.

В структуре *point* задано два параметра: *x* и *y*. Чтобы обратиться, например, к координате *x* точки *A*, требуется написать *Ax*. Аналогично производится обращение к координате *y*.

1.2. *Работа с выражениями и переменными.* Входной язык *LSDSS* поддерживает работу с переменными и выражениями (как числовыми, так и алгебраическими).

Переменные. Рассмотрим объявление и инициализацию переменных, например, *ax* = 100. Данная команда позволяет присвоить переменной *ax* значение 100. Имена переменных не должны содержать заглавных латинских букв, можно использовать строчные латинские буквы, а также цифры.

Выражения. В выражениях можно использовать скобки (), операторы сложения +, умножения *, вычитания -, деления /, возведения в степень **. Также поддерживается функция извлечения квадратного корня *sqrt*(*x*). Рассмотрим пример работы с переменными и выражениями:

A(100, 200)
d = 500
ax = 50
ay = 100

B(*Ax* + *ax*, *Ay* + *d*)
ay = *ax*
C((*Bx* + *ax*) / 2, (*By* + *Ay*) / 2)
e = *sqrt*(*d* + *ay* – 100)
color white
line A B
color red
width 5
line B C

Представляемая версия программы *3D*s-*Geometry* не поддерживает работу с пустыми строками в файле. Также необходимо, чтобы каждая команда была расположена на отдельной строке.

Синтаксис и средства данного языка позволяют создавать стандартные геометрические структуры.

2. Описание алгоритма интерпретации входного файла в программе *3Ds-Geometry*. Задача, рассматриваемая в данной работе, является задачей класса «разбор строк» *в классификации олимпиадных задач по программированию*. Рассмотрим алгоритм.

Программа получает на вход текстовый файл формата *txt* на языке *LSDSS* с кодом для построения изображения. Необходимо обработать и интерпретировать этот файл.

Создается словарь *var* со значениями по умолчанию для операторов *color* (*red*), *width* (1) и *style* (*solid*). В цикле осуществляется считывание файла по строкам, на каждой итерации от текущей линии *line* вызывается функция *com*, служащая для обработки команд, расположенных в заданной строке. Функция *com* определяет, какой оператор находится в текущей строке.

Если это один из «числовых» операторов («числовым» назовем оператор, который изменяет значение какого-либо параметра, за исключением оператора присваивания; числовыми являются операторы *width*, *color*, *dash*, *solid*), то в словаре *var* изменяется текущее значение одного из соответствующих свойств (*color*, *width*, *style*).

Если в данной строке расположен оператор присваивания, то вызывается функция *equal_op*, обрабатывающая выражение, стоящее справа от знака равенства.

Обработка выражения происходит следующим образом. Имена переменных, используемых в заданном выражении, заменяются их значениями, далее вызывается встроенная функция *eval*(*s*), где *s* – строка, позволяющая определить значения заданного числового выражения. Затем в словарь для переменной с соответствующим именем (которое определяется как все, что стоит слева от знака равенства) записывается значение выражения, стоящего справа от знака равенства.

Если в текущей строке задан оператор *line*, то вызывается функция *line_op*. С помощью этой функции проводится линия между заданными точками. Коор-

динаты точек также хранятся в словаре *var*, что позволяет без затруднений в любой момент обратиться к ним или изменить их.

Если в строке нет ни одного из вышеописанных операторов, то в строке задается точка. В таком случае вызывается функция *point_op*. В этой функции в словарь *var* в элемент с индексом, соответствующим имени точки, записываются ее координаты. Также формируется массив так называемых «стационарных» точек – точек, которые должны остаться в плоскости позиционирования в результате построения *3Ds*-изображения. Входной файл обработан.

Далее строится само *3Ds*-изображение, т.е. синяя фигура стереопары. Расстояние между фигурами стереопары задается «ползунками». При нажатии кнопки «*start*» главной формы программы запускается функция *create_3Ds*. В данной функции считываются текущие значения расстояния по *x* и по *y* между фигурами стереопар. Синяя фигура стереопары рисуется аналогично красной фигуре стереопары, но с учетом смещения и «стационарности» точек.

3. Оболочка программы. Для создания *GUI* в *Python* 3.3 использовались библиотеки *tkinter* и *ttk*. При запуске программы открывается главная форма *root = tk.Tk*(). На главной форме расположены виджеты *Label* для создания текста, с помощью оператора *PhotoImage* на главном окне размещаются примеры *3Dss*-изображений, полученных с использованием программы. Также на главной форме представлены поле *Entry* и кнопка *Button*. Поле *Entry* поддерживает *drag&drop* событие. Для работы с *drag&drop* событиями была установлена библиотека *TkDND*. Для работы с этой библиотекой потребовалось создать два дополнительных файла и написать код на *Python* для подключения библиотеки. Входной файл с кодом для изображения перетаскивается в поле *Entry*. Затем, при нажатии кнопки *Start* на окне *root* создается *Canvas*, на котором расположен результат обработки входного файла – *3Ds*-изображение заданной геометрической структуры. На *Canvas* также создаются два «ползунка» – виджеты *Scale*. С помощью этих виджетов регулируется расстояние между красной и синей фигурами стереопары. Для компонентов *Scale* в свойстве *command* указана функция *create_3Ds*, т.е. при изменении значений виджета *Scale* каждый раз происходит прорисовка *3Ds*-изображения, что позволяет настраивать расстояние, наблюдая за изменением картинки. На *Canvas* также расположена кнопка *back*, позволяющая вернуться к главной форме. На главной форме программы располагается основное меню (виджет *tk.Menu*()). С помощью функции *Exit*, находящейся во вкладке *File* основного меню, осуществляется выход из программы (также можно нажать на крестик в правом верхнем углу).

Во вкладке *Help -> Help* расположено окошко с инструкцией по работе с программой. Это окошко содержит виджеты *Label* и кнопку OK, позволяющую закрыть форму. Во вкладке *Help -> About input language* находится форма с подробным описанием синтаксиса входного языка *LSDSS*, а также примерами использования различных операторов языка. На данной форме присутствует полоса прокрутки – виджет *Scrollbar*. Т.к. виджет *Frame* не имеет свойства *yscrollcommand* (именно с помощью этого свойства *Scrollbar* связывается с объектом), то потребовалось создать дополнительную форму, на которой был рас-

положен *Canvas*. К *canvas* был привязан *Scrollbar*, *canvas* находился на вспомогательной форме, которая расположена на форме синтаксиса. Таким образом, появилась возможность прокрутки текста на форме. Также в окошке присутствует кнопка OK, при нажатии которой форма синтаксиса закрывается.

Таким образом, в программе с помощью средств языка *Python* 3.3, встроенных и установленных библиотек был разработан графический интерфейс (*GUI*).

4. Результаты и апробация. Апробация программного продукта ***3Ds-Geometry*** осуществляется учителями математики МОАУ «Гимназия № 8», г. Сочи. В настоящее время программный продукт внедряется в учебный процесс при изучении стереометрии в 10-11 классах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1.http://www.cnews.ru/news/top/2016-08-5_v_rossii_itshnikam_pishut_yazyk_dlya_obshcheniya_s_normalnymi

2. Понятийный и исторический аспекты проблематики разработки 3*Ds*-средств обучения стереометрии в школьном курсе геометрии. // Проблемы теории и практики обучения математике: Сборник научных работ, представленных на Международную научную конференцию «69 Герценовские чтения» / Под ред. В.В. Орлова. СПб.: Издательство РГПУ им. А.И. Герцена, 2016.

А.А. Короткин, Е.П. Кубышкин (Ярославль)

ОБ ОДНОМ ЭФФЕКТИВНОМ МЕТОДЕ КОНТРОЛЯ ЗНАНИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ

Одной из важнейших составляющих образовательного процесса является текущий контроль знаний учащихся. Весьма эффективным видом такого контроля является текущее тестирование, которое может быть автоматизировано на компьютере и проводиться достаточно часто в процессе обучения. Соответствующая программа сама обработает результаты тестирования и выставит оценки в соответствии с заранее заданными критериями, что позволяет легко контролировать весь процесс тестирования. Преподавателю же остается только составить базу данных в виде набора тестов и пронаблюдать, чтобы студенты работали с тестирующей программой самостоятельно. Существует довольно много тестирующих программ. Наиболее удачной из них, на наш взгляд, является программа iTest, являющаяся свободной программируемой оболочкой для подготовки и проведения тестирования [1].Программа состоит из двух частей: клиентской и серверной. Программы iTest позволяют создавать наборы тестов, а так же управлять тестированием учащихся. Программа iTest обладает удобным конструктором тестов. Ответ в тесте может быть как единственным, так и множественным. Вопросы в тесте можно разбить по группам и по уровню сложности. Вопрос может содержать текст, формулу и поясняющее изображение в векторном формате SVG. С клиентской частью работают непосредственно студенты, проходящие тестирование. Клиент может подключаться по локальной сети к серверу с загруженным тестом, или (в случае отсутствия локальной сети) загружать тесты непосредственно на ком-