

## Програмна реализация на средство за обработка и визуализация на данни от учебни сесии

Октай Кър, Ирина Желязкова

### *Program Implementation of a Tool for Processing and Visualization of Learning Session*

**Data:** *The work continues a sequence of papers concerning a teacher's tool for processing and visualization of learning session data. The present paper focuses on program implementation of the tool (choice of the language and environment, class diagram, and file organization).*

**Key words:** *Program Implementation, Programming Language, Class Diagram, File Organization.*

### ВЪВЕДЕНИЕ

В три предишни работи на авторския екип бяха представени UML проектът [1], скриптовия език [2] и алгоритмите на функциониране [3] на средство за таблична обработка и визуализация на данни от учебни сесии (тест, лекция, упражнение). То има за цел да повиши гъвкавостта и продуктивността на работата на преподавателя в сравнение с използването на EXCEL.

Настоящият материал фокусира върху програмната реализация на това средство, което представлява алтернатива на EXCEL. В следващия параграф се аргументира изборът на езика и средата за програмиране. В третия параграф се коментира диаграмата на класовете, а в последния е дадена файловата организация на данните. Заключениеето съдържа основните резултати и бъдещи планове на колектива.

### ИЗБОР ЕЗИК И СРЕДА ЗА ПРОГРАМИРАНЕ

Като процедурен и обектно-ориентиран език за програмната реализация на средството бе избран **C++**. Той има по-малко ключови думи в сравнение с други процедурни езици. Отнася се към езиците от средно ниво като комбинира и елементи на асемблерните езици, т. е. позволява директна обработка на битове, байтове и адреси. Друга особеност е, че кодът на C++ е преносим, което означава, че софтуер, написан на един вид микропроцесори, лесно се приспособява да работи на друг.

Като среда за програмиране бе предпочетена **Borland C++ Builder 6**. Тя предлага стандартна библиотека с голям брой функции, осигуряващи изпълнение на задачи за обработка на стрингове и файлове. Освен библиотеката с визуални компоненти (**Visual Component Library - VCL**), от средата е достъпна е мощната библиотека **Vector**. Тя е особено подходяща за лесно и бързо създаване и използване на различни динамични структури от данни като списъци, масиви, множества и др.

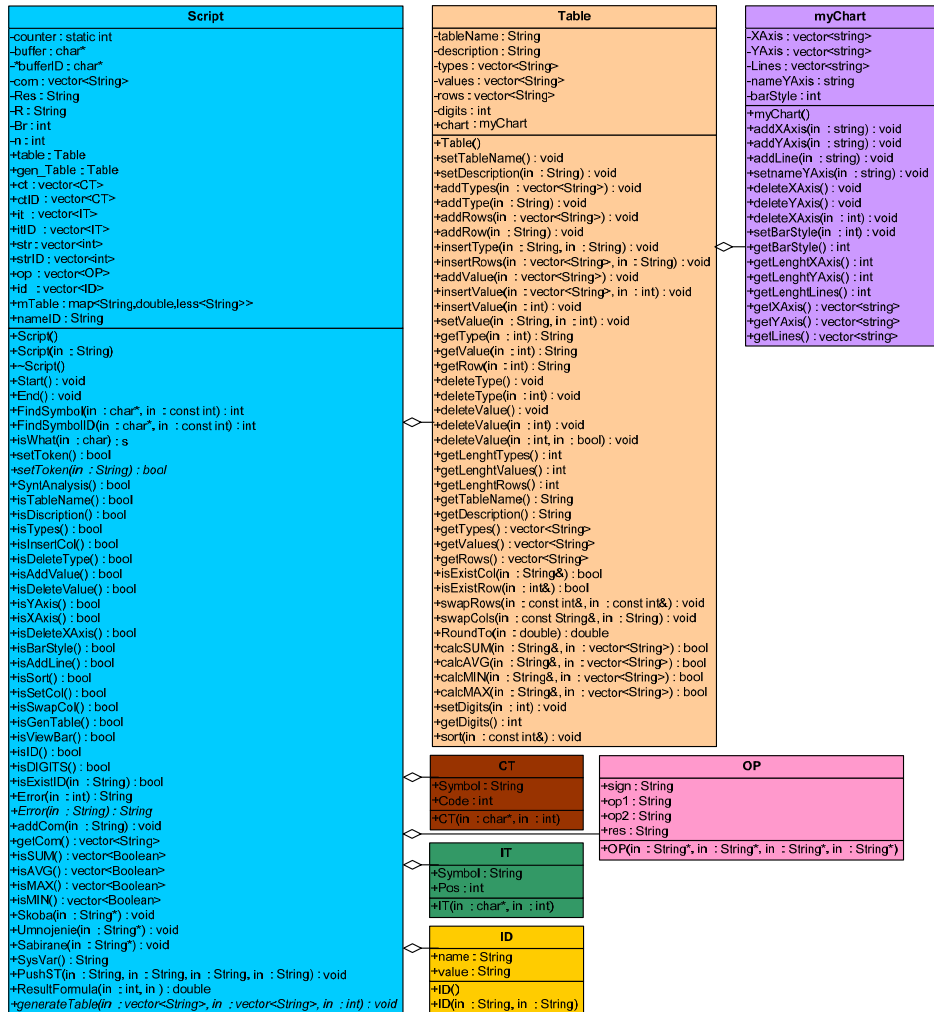
### ЙЕРАРХИЯ НА КЛАСОВЕТЕ

Йерархията на класовете, добавени към системните, е представена на фиг. 1. Общият им брой е 7, като основни от тях са три (**Script**, **Table** и **MyChart**), а останалите четири (**Struct CT**, **Struct IT**, **Struct OP**, **Struct ID**) са спомагателни. Описанието на основния клас **Script** тук не е дадено поради големия му обем. Типът, видимостта и предназначението на атрибутите на останалите класове (**Table**, **MyChart**, **CT**, **IT**, **OP** и **ID**) както и предаваните параметри, резултатът, видимостта и флагът за константа на техните методи са сведени в табл. 1,2,3,4,5,6.

### ФАЙЛОВА ОРГАНИЗАЦИЯ

Файловата организация на програмните модули и данните е показана на фиг. 3. Освен файла на проекта (**Project1.cpp**) има още шест **.cpp** файла (**Unit1**, **Unit2**, **Unit3**, **Script**, **Table**, **myChart**), шест **.h** файла (**Unit1**, **Unit2**, **Unit3**, **Script**,

Table, myChart) и шест файлове с данни (.tbl, .gtbl, .abc). На същата фигура са показани и връзките от тип include между .h и .cpp файловете.



Фиг. 1. Йерархията на класовете

Табл. 1. Описание на атрибутите и методите на класа **Table**

Атрибут	Видимост	Предназначение	
tableName	private	Брояч на променливи от този клас	
description	private	Описание на таблицата	
types	private	Едномерен вектор от тип String за имената на колоните	
values	private	Едномерен вектор от тип String за стойностите на клетките	
rows	private	Едномерен вектор от тип String за имената на редовете	
digits	private	Брой на цифрите след десетичната точка в стойностите	
chart	public	Променлива за бар диаграма на таблицата	
Функция	Видимост	Константа	Предназначение
Table	public	-	Конструктор
setTableName	public	-	Задава име на таблицата
setDescription	public	-	Задава описание на таблицата
addTypes	public	-	Добавя нови колони на таблицата
addType	public	-	Добавя нова колона на таблицата
addRows	public	-	Добавя нови редове на таблицата и задава имената на редовете
addRow	public	-	Добавя нов ред на таблицата и задава името на реда
insertType	public	-	Вмъква нова колона в таблицата
insertRows	public	-	Задава стойности на клетките по избрана колона
addValue	public	-	Добавя нов празен ред на таблицата
insertValue	public	-	Вмъква нов ред в таблицата и задава стойности на клетките
insertValue	public	-	Вмъква нов празен ред в таблицата
setValue	public	-	Задава стойност на избрана клетка
getType	public	√	Връща името на избрана колона
getValue	public	√	Връща стойност за избрана клетка
getRow	public	√	Връща името на избран ред
deleteType	public	-	Изтрива всички колони от таблицата
deleteType	public	-	Изтрива избрана колона от таблицата
deleteValue	public	-	Изтрива всички редове от таблицата
deleteValue	public	-	Изтрива избран ред от таблицата
deleteValue	public	-	Изтрива клетките на избрана колона
getLenghtTypes	public	√	Връща брой колони от таблицата
getLenghtValues	public	√	Връща брой елементи от списъка Values
getLenghtRows	public	√	Връща брой елементи от списъка Rows
getTableName	public	√	Връща името на таблицата
getDescription	public	√	Връща обяснението на таблицата
getTypes	public	√	Връща вектор от списъка Types
getValues	public	√	Връща вектор от списъка Values
getRows	public	-	Връща вектор от списъка Rows
isExistCol	public	√	Проверява вектор Types дали включва зададен параметър от тип String, ако да връща true , а ако не връща false
isExistRow	public	√	Проверява има ли таблицата толкова реда колкото е зададената стойност
swapRows	public	-	Размества два съседни реда
swapCols	public	-	Размества две съседни колони
sort	public	-	Сортира клетките по зададена колона
RoundTo	public	-	Закръгля до указания брой знаци
calcSUM	public	-	Изчислява сума по хоризонтала
calcAVG	public	-	Изчислява средно аритметично
calcMIN	public	-	Намира минимална стойност
calcMAX	public	-	Намира максимална стойност
setDigits	public	-	Задава стойността на digits
getDigits	public	-	Връща стойността на digits

Табл. 2. Описание на атрибутите и методите на класа **myChart**

Атрибут	Видимост	Предназначение	
XAxis	private	Стойности от тип string на баровете по координатната ос X	
YAxis	private	Стойности от тип double на баровете по координатната ос Y	
Lines	private	Стойност от тип double по координатната ос Y на хоризонтална линия	
nameYAxis	private	Стойност от тип string по координатната ос Y на хоризонтална линия	
barStyle	private	Стойност от тип int за различните видове диаграми	
Функция	Видимост	Константа	Предназначение
myChart	public	-	Конструктор
addXAxis	public	-	Добавя нов елемент в спътника XAxis
addYAxis	public	-	Добавя нов елемент в спътника YAxis
addLine	public	-	Добавя нов елемент в спътника Lines
setNameYAxis	public	-	Задава името на променливата YAxis

getXAxis	public	√	Връща вектор от списъка XAxis
getYAxis	public	√	Връща вектор от списъка YAxis
getLines	public	√	Връща вектор от списъка Lines
getNameYAxis	public	√	Връща името на променливата YAxis
deleteYAxis	public	-	Изтрива стойност на променливата YAxis
deleteXAxis	public	-	Изтрива всички стойности от променливата XAxis
deleteXAxis	public	-	Изтрива стойност по зададен индекс в списъка XAxis
setBarStyle	public	-	Задава вида за изчертаване на chart
getBarStyle	public	√	Връща променливата barStyle от този клас
getLenghtXAxis	public	√	Връща брой елементи от списъка XAxis
getLenghtYAxis	public	√	Връща брой елементи от списъка YAxis
getLenghtLines	public	√	Връща брой елементи от списъка Lines

Табл. 3. Описание на атрибутите и методите на класа **Struct CT**

Атрибут		Предназначение	
Symbol		Имената на лексикалните единици (идентификатори, или служебни думи, числови константи, операционни символи, разделители)	
Code		Съхранява атрибута на съответния символ	
Функция	Видимост	Константа	Предназначение
CT	public	-	Конструктор

Табл. 4. Описание на атрибутите и методите на класа **Struct IT**

Атрибут		Предназначение	
Symbol	public	Имената на идентификатори или служебни думи, числови константи, операционни символи или разделители, фигури	
Pos	public	Указател към позиция на Symbol	
Функция	Видимост	Константа	Предназначение
IT	public	-	Конструктор

Табл. 5. Описание на атрибутите и методите на класа **Struct OP**

Атрибут		Предназначение	
sign	public	Код на операцията	
op1	public	Адреси на операнди или номера на тетради	
op2	public	Адреси на операнди или номера на тетради	
res	public	Вътрешна променлива за съхраняване на междинните резултати	
Функция	Видимост	Константа	Предназначение
OP	public	-	Конструктор

Табл. 6. Описание на атрибутите и методите на класа **Struct ID**

Атрибут		Предназначение	
name	public	Име на идентификатор на променливата от лявата страна на израза	
value	public	Стойност на идентификатор на променливата от дясната страна	
Функция	Видимост	Константа	Предназначение
ID	public	-	Конструктор
ID	public	-	Конструктор

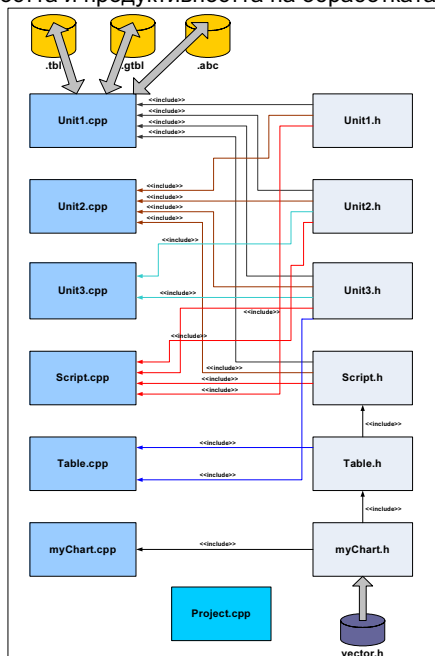
Съдържанието на отделните файлове е както следва:

- **Project1** - входната точка на програмата; **Unit1** – главното меню на програмата;
- **Unit2** – прозорец за въвеждане на команди; **Unit3** – прозорец за визуализиране на таблицата;
- **Unit1, Unit2, Unit3** – деклариране на функции и променливи на компонентите съответно на класа **TForm1, TForm2** и **TForm3**;
- **Script, Table, myChart** – деклариране на едноименния клас, съдържащ функциите и променливите съответно на скрипта, таблицата и бар диаграмата.
- **.tbl** – данните на таблицата и данните на бар диаграмата;
- **.gtbl** – данните на генерирана таблица и на мулти бар диаграмата;

- **.abc** – съдържа командите, написани в прозореца на текстовия редактор.

### ЗАКЛЮЧЕНИЕ

Представена е програмната реализация на средство за обработка и визуализация на данни от учебни сесии (тест, лекция, упражнение) в средата **Borland C++ Builder 6** с използването на обектно-ориентиран стил на програмиране. Средството може да се разглежда като алтернатива на досегашната обработка с **EXCEL** [5]. Разликата е, че в съответствие със стандартизирана методика преподавателят програмира на специализиран скриптов език с отворен код, което повишава гъвкавостта и продуктивността на обработката.



Фиг. 2. Файлова организация

### ЛИТЕРАТУРА

- [1] Кър О., Желязкова И., UML проект на средство за обработка и визуализация на данни от учебни сесии, Трудове на научната сесия на Русенския университет, 2009.
- [2] Кър О., Желязкова И., Скриптов език за обработка и визуализация на данни от учебни сесии, Трудове на научната конференция на Русенския университет, 2009.
- [3] Кър О., Желязкова И., Алгоритми за функциониране на средство за обработка и визуализация на данни от учебни сесии, Трудове на научната сесия на Русенския университет, 2009.
- [4] Тодорова М., Програмиране на C++, част 1, София, "Ciela", 2004.
- [5] Zheliazkova I. I., Kolev R. T., Task Results Processing for the Needs of Task-Oriented Design Environments, *Int. J. Computers & Education*, vol. 51, 2008, pp. 86-96.

### За контакти:

Доц. д-р Ирина Желязкова, Катедра "Компютърни системи и технологии", Русенски университет "Ангел Кънчев", тел.: 082-888 711, e-mail: [Irina@ecs.ru.acad.bg](mailto:Irina@ecs.ru.acad.bg)

Докладът е рецензиран.