Elżbieta Jasińska*, Jan Ruchel*

Using a Spreadsheet for Surveying Computations**

1. Visual Basic for Applications – Easy and Effective Programming Language

The computations related to surveying works require a proper application guaranteeing the execution of the tasks. We have a choice therefore either to purchase a program that can execute our computations or to create a new application from the ground up. Or we can make use of a standard calculating program (e.g. a spreadsheet), which can provide capabilities to define all details of the task realization as well as the method and scope of presenting the results.

A properly designed spreadsheet becomes an application that can realize various tasks from the simplest computations to complex analyses. On our decision depends what functions the application can execute.

The commercial programs available nowadays at the market usually present the final results only, without providing any view into indirect results of computations – which are often indispensable for executing the analysis, especially in case of any computation problems. While making our own application in a form of a spreadsheet realizing the wanted task we can ensure the access to the indirect results at any stage of the computations.

There are many ways for creating an application capable of performing the computations we need:

- Creating a software from the ground up, using some existing programming tools (for example C++, Delphi, Java, Visual Basic or others).
- Using the popular Excel spreadsheet program. Apart from its wide calculating capabilities (including the matrix algebra) the software features embedded elements for constructing macros with the use of Visual Basic for Applications (VBA). With these capabilities it allows to create a complete application sufficient for a given scope of computation tasks.

* AGH University of Science and Technology, Faculty of Mining Surveying and Environmental Engineering, Department of Geomatics, Krakow
** The paper has been worked out within University research program no. 11.1.150.006

Below we provide an exemplary procedure of employing the Excel spreadsheet and VBA language for creation of a computing application capable of the adjustment of a classical leveling net.

In the first years of development of informatics the programmers were regarded as immensely talented individuals and the programming as inaccessible for common people. Along with the hardware development the programming languages were also evolving, starting from a low-level ones (assemblers) to the high-level languages such as C++. At present they are less complicated, feature many mechanisms making the programming easier and they became more accessible for common computer users. The BASIC language was created in 1963 as a very simple programming language designed for beginning programmers. The breakthrough came with Visual Basic 1.0. It appeared at the time of releasing the Microsoft Windows operating system. In this version it was already a visual language in which a great part of programming relied on graphical arrangement of objects in the application window (called a form). There was no need any more to construct very enlarged source codes in which most of the programming work was employed to create a shape of the application. In Visual Basic a programmer can quickly create fully working application with the help of a graphical user interface (IDE). **Visual Basic for Applications** (VBA) is a Visual Basic based programming language implemented into Microsoft Office applications and a few others (e.g. AutoCAD). This version of Visual Basic serves primarily to automatize the work with documents through the use of macros. The main difference between VBA and VB is that VBA does not allow to create independent complex applications of the EXE type. The code written in VBA is always enclosed in the document created with a program capable of using the VBA – for example in the *.XLS file of the MS Excel spreadsheet. A program of this type therefore requires a runtime environment, such as an application supporting the given document format, installed on the programmer's computer. Since version 2000 the MS Office package is furnished with a separate VB editor – the well known for programmers VB 6.0. This feature makes the coding work much easier.

2. Creating an Event-Driven Application (Program Form and Code)

An application written in Visual Basic language consists of two elements:

A. Graphical user interface, or a window system called a form, and objects assigned to it. In easy tasks singular objects – most commonly buttons – can be used instead of a form.
B. Code of the program, containing procedures with a set of instructions to be executed.

Applications written in VB or VBA are event-based programs, what means the proper procedure is triggered when a certain event occurs. By an event one should understand an activity made by the user, such as clicking the mouse button, pressing a proper key on the keyboard, marking with the cursor, etc. Events can be also linked with some peripherals (for example measuring instruments) or with a certain time periods. Occurring or not occurring of the wanted event determines the sequence of the code executed. This is why during programming we should take care of the proper sequence of events. In creating applications one general rule should be primarily observed – do not empower the user to make an error, for if you do they will certainly make it.

The event has to be linked to a particular object from the user interface within our application if it is to trigger a proper reaction such as executing the instructions.

The procedure of handling an event has the following syntax:

Private Sub Object_event (optional parameters of the procedure)
.....
'a list of instructions assigned to the object and the event
.....
End Sub

As it can be seen, the name of the procedure contains the name of the object and the event associated with it. It can also include certain parameters proper for the type of objects and events involved. The headers of the procedures are automatically generated by VBA.

Procedure is a very important component of the program, because for the code to be executed it must be enclosed into a procedure. A procedure makes the smallest part of the code which can be executed independently from other parts of the code.

Module contains one or more procedures and a declaration section in which declarations common for all the procedures for the given module are placed.

Project consists of all the modules, forms and objects of the home application of the document being created and the document itself.

VBA has a very complex (multi-window) and at the same time user friendly interface, providing convenient ways of creating applications. The most important windows are:

- Project Explorer window, containing all elements of the spreadsheet and of the project.
- View Designer window, which allows to design the look of our form in the graphical mode.
- Toolbox window, containing the list of available objects that can be used in the projecting mode.

– Properties window, displaying present values of all the properties for the object active in the Project Explorer. In this window we can modify the properties of the active object. If needed, the values can be changed also from the code level in time of executing certain procedures.
– Code window, containing the procedures and declarations created for the project.

3. Writing the Event Procedures

Writing the procedures is quite easy. To write an event procedure for a spreadsheet do the following:

– Run the Visual Basic Editor.
– If the Project window is not displayed, call it from the View menu choosing option ProjectExplorer.
– In the Project window double-click on the name of the spreadsheet you want to create the event procedure for. This should display the Code window for this spreadsheet.
– In the Code window at the Object drop-down list (top left part of the window) choose the Worksheet object. After choosing the object MS Excel will automatically create a template for default event procedure, proper for the given object type. The default event for the worksheet is the SelectionChange event.
– We can also choose an event other than default from the Procedure drop--down list (top right part of the Code window). The list shows all possible events associated with the chosen one. After selecting a certain event the proper template for the event procedure will be created (Fig. 1).

Adding objects to a worksheet is done by placing them on the sheet with the mouse cursor and defining their size and position with the mouse and the mechanisms of the graphic environment of MS Windows. With the Properties window we can determine values for the chosen object's characteristics (Fig. 2).

Visual Basic for Applications is a user friendly programming language on account of the embedded error control feature. The errors occurring during writing a program can be divided into several groups:

– Text editing errors – these can be avoided by a careful examination of what has been written. These errors are not dangerous for the application.
– Errors occurring in time of compilation – making the compilation of the program impossible – such as: syntax mismatches for declarations and instructions, or wrong identification of certain program elements.

– Execution errors – occurring while running a part of algorithm. To prevent this kind of errors certain safety mechanisms (error traps) are applied while writing the code.
– Logic errors – the most dangerous kind of errors, for they evoke wrong computation results. They are also the most difficult to find, because no messages signalize them. Usually logic errors come from executing a wrongly written algorithm.

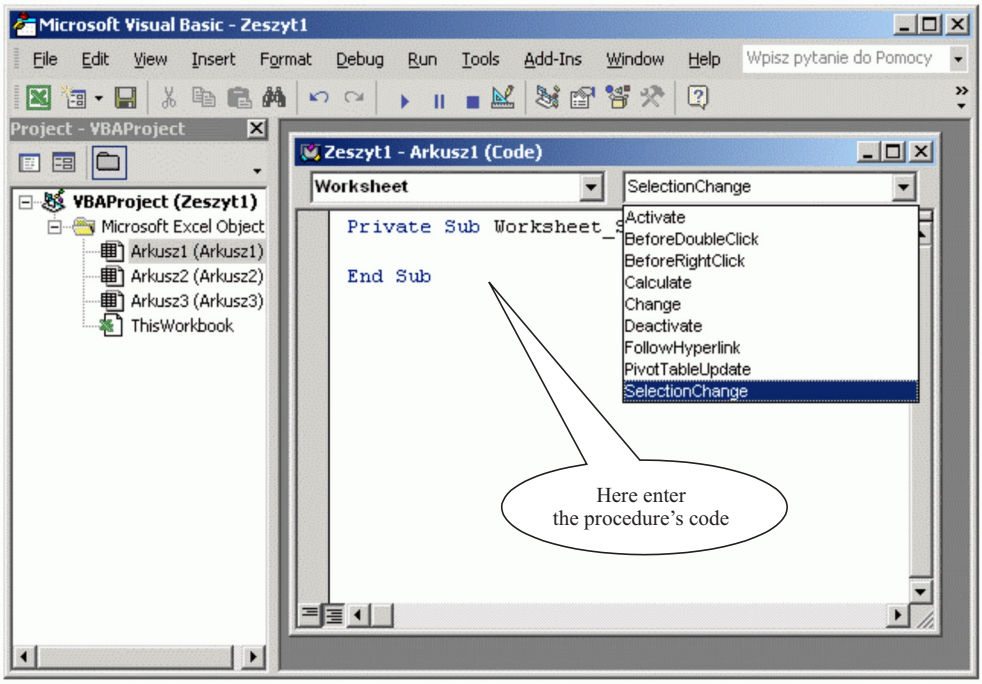


Fig. 1. The program code window – an event procedure template (Excel 2003)

Before the application is written the requirements of the program need to be carefully determined and the algorithm leading to wanted results defined and checked. Preparing the application to computations requires designing (and reserving proper memory resources for them) certain structures in which our data will be stored. The structure consists mainly of arrays and records, structural types containing many elements. Arrays contain elements of one type only (but this can be also a structural type), records may contain elements of various types. Creating these structures is called declaration. In case of arrays we can use dynamic declaration, i.e. change the array's size accordingly to needs.

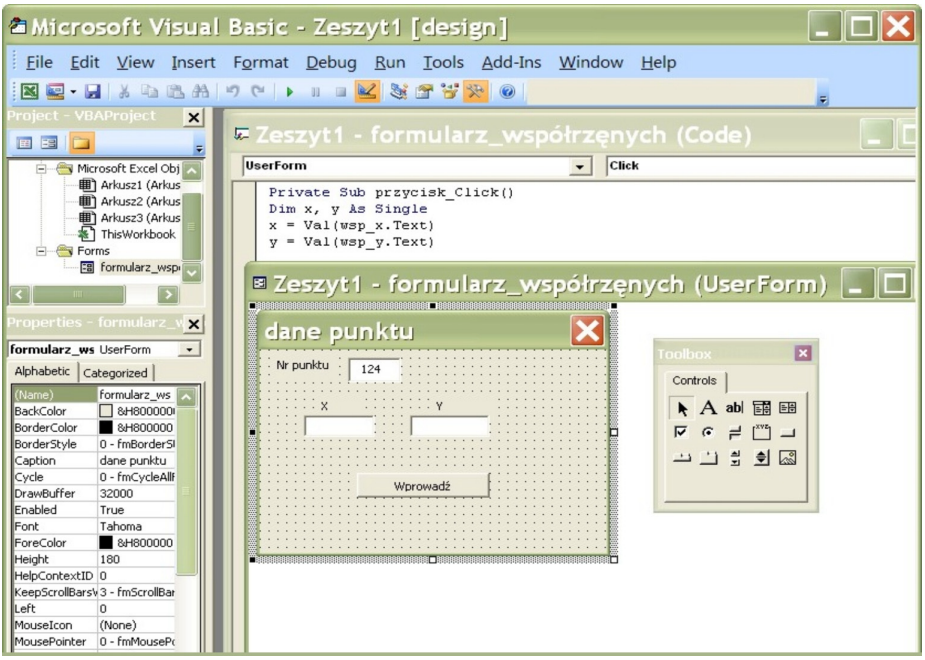


Fig. 2. The program code window – creating a worksheet (Excel 2007)

Declaring of an array with the elements' type only determined:

Dim matrixA () As Single

This is a two-pointer array containing elements of the real type with a single precision. When the array is going to be used we can predeclare it along with determining the size and number of pointers:

ReDim matrixA (w, k)

as a new matrix, or:

ReDim Preserve matrixA (w, k)

as a matrix with a different size but the content unaltered.

3.1. An Exemplary Application for Adjusting the Classical Leveling Net (within the Basic Scope)

Basic programming knowledge is enough to make use of VBA to create an application capable of running any process of surveying computations. Of course in order to create such application the knowledge concerning the task itself is also required; one must know the algorithm resolving the problem.

The illustration applied for this paper will be a computation of the classical leveling net according to the adjustment algorithm by means of the least squares method.

The basic formulas used in the program are listed below:

– Observation weights:

$$p_i = \frac{1}{m_i^2}.$$

– The system of correction equations:

• for linear observations:

$$w_i = dD_i + D_{approx.} - D_{measured},$$

where:

$$dD_i = \begin{vmatrix} dx_p & dy_p & dx_k & dy_k \\ -\cos(A_{p-k}) & -\sin(A_{p-k}) & \cos(A_{p-k}) & \sin(A_{p-k}) \end{vmatrix}_2,$$

p – starting point,
k – ending point,
D_i – observation measured,
w_i – distance correction,
$D_{approx.} = \sqrt{\Delta x^2 + \Delta y^2}$ – approximated distance
A_{p-k} – azimuth for the side measured
dx_p, dx_k, dy_p, dy_k – partial coordinates;

• for angular observations:

$$v_i = d\alpha_i + \alpha_{approx.} - \alpha_{measured},$$

where:

$$d\alpha_i = \begin{vmatrix} dx_l & dy_l & dx_p & dy_p & dx_c & dy_c \\ A_l & B_l & -A_p & -B_p & -(A_l - A_p) & -(B_l - B_p) \end{vmatrix}_1,$$

α_i – observation measured,
v_i – angle correction,
α_i^0 – approximated angle,
A_l, B_l, A_p, B_p – gradients respectively for the left and right arm of the angle:

$$A = \frac{\Delta x}{\Delta x^2 + \Delta y^2} \cdot \rho, \quad B = \frac{\Delta y}{\Delta x^2 + \Delta y^2} \cdot \rho,$$

dx_l, dx_c, dx_p, dy_l, dy_c, dy_p – partial coordinates.

– The correction equations in the matrix form:

$$V = A \cdot x + w.$$

– The solution of the system of standard equations:

$$(A^T \cdot p \cdot A) \cdot X + (A^T \cdot p \cdot w) = 0,$$
$$-x = (A^T \cdot p \cdot A)^{-1} \cdot (A^T \cdot p \cdot w).$$

– Computation of adjusted coordinates:

$$X_i = X_i^0 + dx_i,$$
$$Y_i = Y_i^0 + dy_i.$$

– Basic analysis of accuracy:

$$m_0 = \pm\sqrt{\frac{[pvv]}{n-u}}, \qquad \mathrm{cov}(x) = m_0^2 \cdot (A^T \cdot p \cdot A)^{-1}.$$

The program code enables to display the results on the monitor in the desired form and to save them into files (e.g. text files).

Exemplary windows from the application thus created (Figs 3–6).

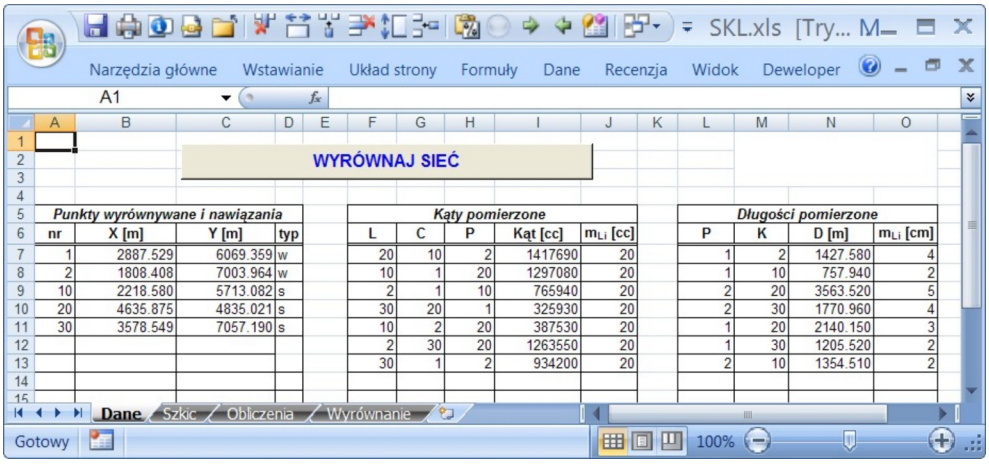


Fig. 3. The start window of the finished application designed for adjusting the classical leveling net – it requires entering data and running by pressing the WYRÓWNAJ ('adjust') button

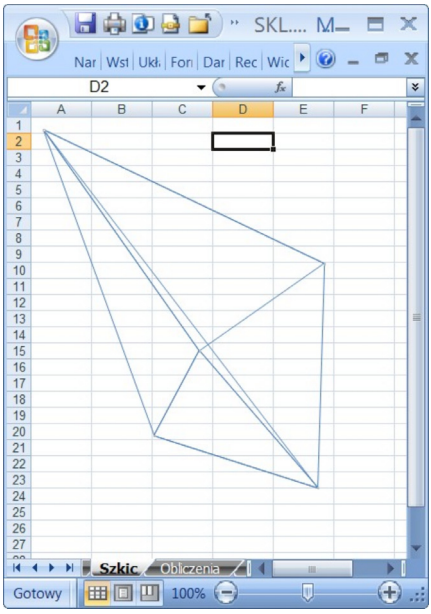
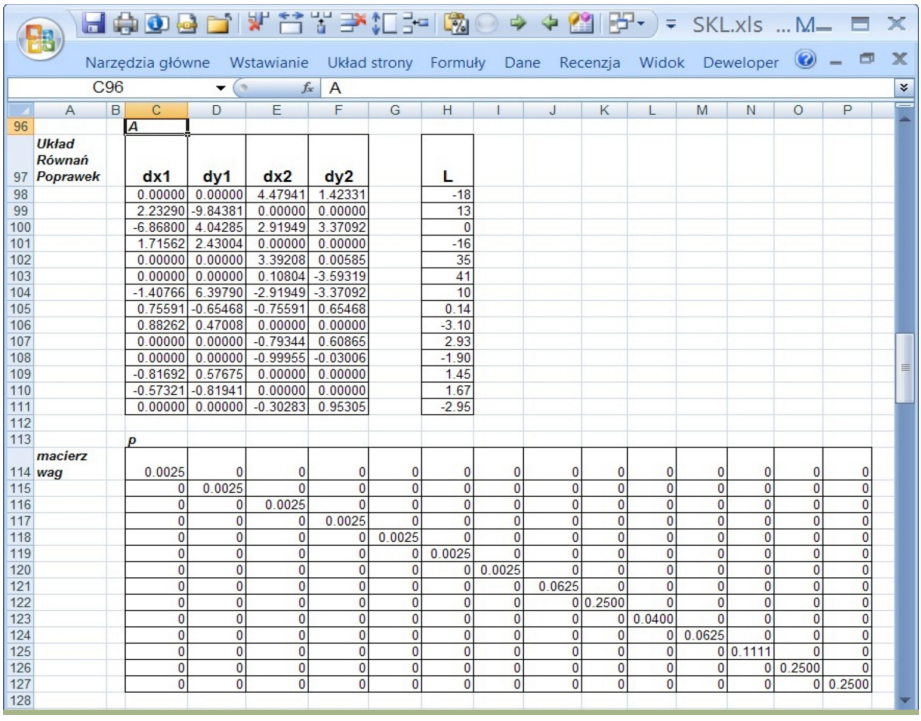
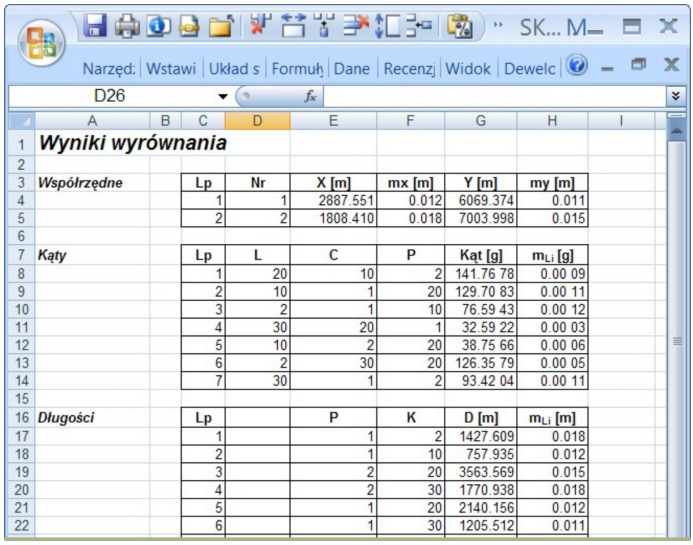


Fig. 4. A worksheet with a draft of the net adjusted



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
96			*A*													
97	*Układ Równań Poprawek*		**dx1**	**dy1**	**dx2**	**dy2**		**L**								
98			0.00000	0.00000	4.47941	1.42331		-18								
99			2.23290	-9.84381	0.00000	0.00000		13								
100			-6.86800	4.04285	2.91949	3.37092		0								
101			1.71562	2.43004	0.00000	0.00000		-16								
102			0.00000	0.00000	3.39208	0.00585		35								
103			0.00000	0.00000	0.10804	-3.59319		41								
104			-1.40766	6.39790	-2.91949	-3.37092		10								
105			0.75591	-0.65468	-0.75591	0.65468		0.14								
106			0.88262	0.47008	0.00000	0.00000		-3.10								
107			0.00000	0.00000	-0.79344	0.60865		2.93								
108			0.00000	0.00000	-0.99955	-0.03006		-1.90								
109			-0.81692	0.57675	0.00000	0.00000		1.45								
110			-0.57321	-0.81941	0.00000	0.00000		1.67								
111			0.00000	0.00000	-0.30283	0.95305		-2.95								
112																
113			*p*													
114	*macierz wag*		0.0025	0	0	0	0	0	0	0	0	0	0	0	0	0
115			0	0.0025	0	0	0	0	0	0	0	0	0	0	0	0
116			0	0	0.0025	0	0	0	0	0	0	0	0	0	0	0
117			0	0	0	0.0025	0	0	0	0	0	0	0	0	0	0
118			0	0	0	0	0.0025	0	0	0	0	0	0	0	0	0
119			0	0	0	0	0	0.0025	0	0	0	0	0	0	0	0
120			0	0	0	0	0	0	0.0025	0	0	0	0	0	0	0
121			0	0	0	0	0	0	0	0.0625	0	0	0	0	0	0
122			0	0	0	0	0	0	0	0	0.2500	0	0	0	0	0
123			0	0	0	0	0	0	0	0	0	0.0400	0	0	0	0
124			0	0	0	0	0	0	0	0	0	0	0.0625	0	0	0
125			0	0	0	0	0	0	0	0	0	0	0	0.1111	0	0
126			0	0	0	0	0	0	0	0	0	0	0	0	0.2500	0
127			0	0	0	0	0	0	0	0	0	0	0	0	0	0.2500
128																

Fig. 5. A worksheet with computations (a fragment) of the net adjusted



Wyniki wyrównania

Współrzędne	Lp	Nr	X [m]	mx [m]	Y [m]	my [m]
	1	1	2887.551	0.012	6069.374	0.011
	2	2	1808.410	0.018	7003.998	0.015

Kąty	Lp	L	C	P	Kąt [g]	m_{Li} [g]
	1	20	10	2	141.76 78	0.00 09
	2	10	1	20	129.70 83	0.00 11
	3	2	1	10	76.59 43	0.00 12
	4	30	20	1	32.59 22	0.00 03
	5	10	2	20	38.75 66	0.00 06
	6	2	30	20	126.35 79	0.00 05
	7	30	1	2	93.42 04	0.00 11

Długości	Lp		P	K	D [m]	m_{Li} [m]
	1		1	2	1427.609	0.018
	2		1	10	757.935	0.012
	3		2	20	3563.569	0.015
	4		2	30	1770.938	0.018
	5		1	20	2140.156	0.012
	6		1	30	1205.512	0.011

Fig. 6. The result window with results of the leveling net adjustment computations

4. Summary

Choosing the Visual Basic for Applications to create an application provides many advantages. The project can be expanded (new features added) with no limitations. It provides access to the indirect results and gives possibility to display the results in any desired form. We bear no additional costs – assuming we own the MS Office application packet. An element of a very high importance here is a proper decimal separator character (for a given Windows installation). Visual Basic for Applications (and Visual Basic) requires the "." (dot) as a decimal separator. In the Windows systems changing the decimal separator can be made through the Control Panel (regional settings or international settings – depending on a system version).

The text files created during the computations are very useful as they are easy to view and edit. They can also be employed in other programs. In particular, the cooperation with AutoCad becomes convenient and easy, as the program also has the implementation of VBA language embedded.

References

[1] Czaja J.: *Modele statystyczne w informacji o terenie*. Wydawnictwa AGH, Kraków 1996.

[2] Hausbrandt S.: *Rachunek wyrównawczy i obliczenia geodezyjne. Vol. 1 and 2*. PPWK, Warszawa 1971.

[3] Hyde R.: *Profesjonalne programowanie. Część 1: Zrozumieć komputer*. Helion, Gliwice 2005.
[4] Hyde R.: *Profesjonalne programowanie. Część 2: Myśl niskopoziomowo, pisz wysokopoziomowo*. Helion, Gliwice 2006.
[5] *Instrukcja techniczna O-1: Ogólne zasady wykonywania prac geodezyjnych*. Warszawa 1988.
[6] Łoś A.: *Rachunek wyrównawczy.* PWN, Warszawa 1973.
[7] Mysior M.: *20 lekcji z Visual Basic dla aplikacji. Wprowadzenie do programowania*. Mikom, Warszawa 2002.
[8] Roman S., Petrusha R., Lomax P.: *VB.NET Almanach*. Helion, Gliwic 2003.
[9] Walkenbach J.: *Excel 2003 PL. Programowanie w VBA. Vademecum profesjonalisty*. Helion, Gliwice 2004.
[10] Willet E.C., Cummings S.: *ABC Visual Basic dla aplikacji w Office XP*. Helion, Gliwice 2002.