TEST MODULE OF A SOFTWARE SYSTEM FOR SYNTHESIS AND ANALYSIS OF COMMUNICATION CIRCUITS AND PROCESSES

Adriana Naydenova Borodzhieva[1], *Tanya Metodieva Stoyanova*[2]

[1]"Telecommunications" Department,
[2]"Theoretical and Measuring Electrical Engineering" Department,
University of Ruse "Angel Kanchev"

Bulgaria, 7017 Ruse, 8 "Studentska" Street, tel.: [1](00359 82) 888 734, [2](00359 82) 888 502
e-mail: [1]aborodjieva@ecs.uni-ruse.bg, [2]tstoyanova@uni-ruse.bg

Abstract: *In this paper a test module of a software system for synthesis and analysis of communication circuits and processes is described. It is used in the educational process in the course "Communication Circuits", included as compulsory in the curriculum of the specialty "Telecommunication Systems" for the Bachelor educational degree. The test module of the software system, developed by MATLAB, offers the opportunity to test the students in the following sections of material: oscillating circuits, electrical filters and modulations.*

Keywords: *test module, communication circuits and processes, oscillating circuits, electrical filters, modulations.*

1. INTRODUCTION

According to the current syllabus for the course "Communication Circuits" included as mandatory in the curriculum of the specialty "Telecommunication Systems" for the Bachelor educational degree at the University of Ruse "Angel Kanchev", during the practical exercises students solve individual problems in accordance with certain methods [1, 2]. The problems solved should be submitted to the teacher at the end of the exercise.

A test module of a software system developed for synthesis and analysis of communication circuits and processes is considered in the paper.

2. TEST MODULE – IMPLEMENTATION

The test module of the system is designed to assess the knowledge and skills of students allowing the teacher to form the final marks for a problem, for an exercise or for the term (for each student).

The test module is based on the use of both functions developed: *test* and *mark*, whose flow-charts are shown in Fig. 1 and Fig. 2.

Testing students is realized using the function *test*. The calculation module of the system enables the calculation of various parameters and the test module generates 5 possible values for the quantity requested (this number can be increased, which will reduce the chance of guessing the correct answer). The student chooses one of these five options the closest one to the value calculated by him/her. When the correct answer for the quantity calculated the function *test* adds a fixed number of points. When the answer is wrong, the function *test* takes off a fixed number of points. By the withdrawal of points when wrong answers the willingness of students to “blindly guessing the correct answer” is reduced.

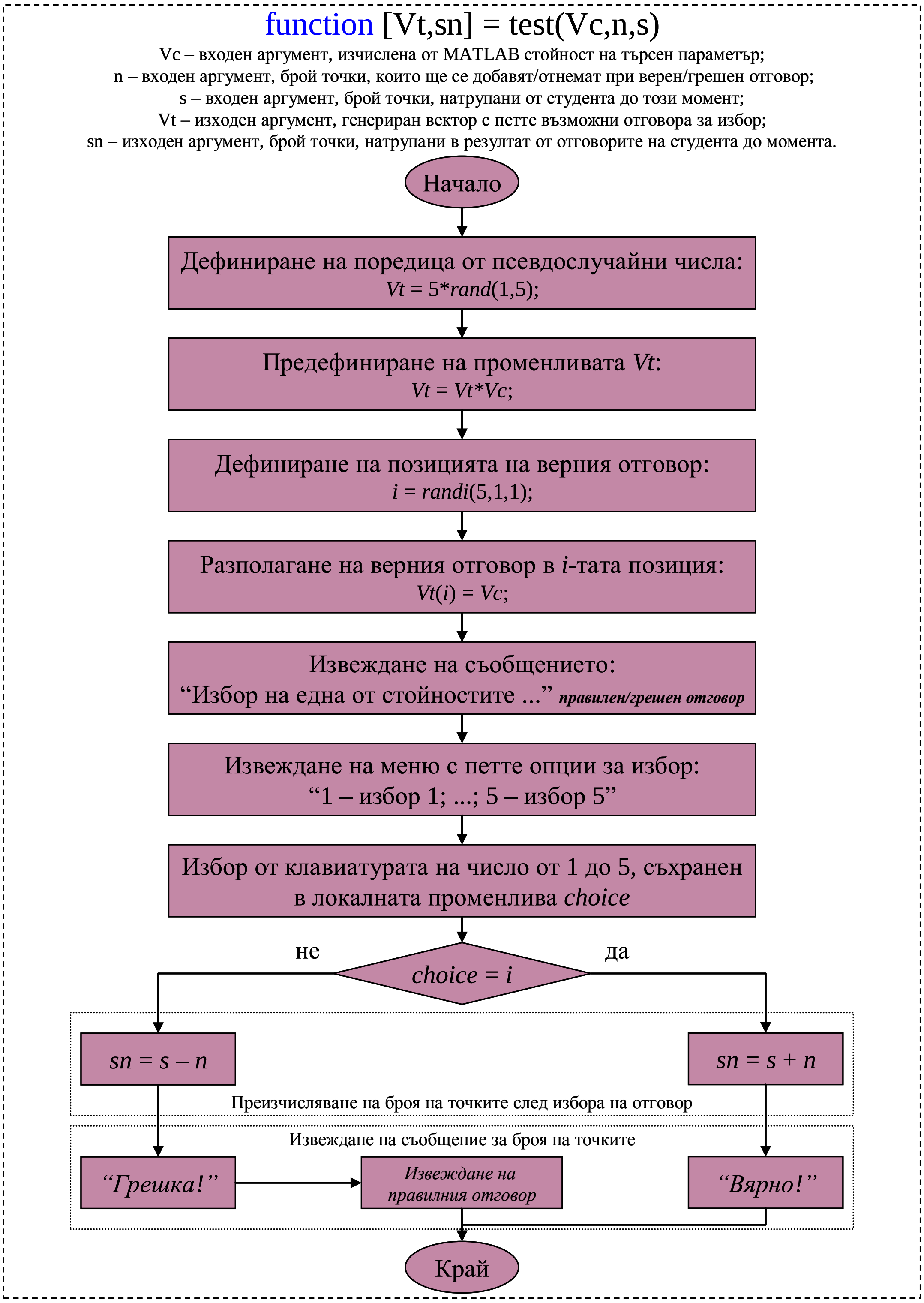

Fig. 1. Block diagram of the algorithm of the function *test*
for generating five different options for selecting the correct answer

The test module is tested at calculated "exact" value 100.0000. 20 trials are done and the results of testing are presented in Table 1.

Using the built-in MATLAB functions *tic* and *toc* the time required for generating the five possible values for answers, for generating the position the correct answer will be located at and for the settlement can be determined. The simulation time (in seconds) is indicated in the last column of Table 1 – it is of the order of 4 – 8 *ms*, i.e. too little.

Table 1. Simulation results of the testing the function *test*

№	***i***	**Select "1 of 5" for a value of *R***					**Time, *sec***
1	**2**	289.8523	**100.0000**	72.4774	426.5156	311.0276	0.005331
2	**1**	**100.0000**	200.9040	37.9833	119.9581	61.6595	0.004993
3	**3**	119.9763	208.6335	**100.0000**	451.3581	472.3936	0.005270
4	**4**	244.6263	168.8597	450.0269	**100.0000**	55.6014	0.005276
5	**5**	194.8694	120.8456	201.9561	48.2273	**100.0000**	0.005551
6	**5**	478.0673	287.6043	29.8898	117.3900	**100.0000**	0.005859
7	**4**	7.7017	21.5119	84.4950	**100.0000**	365.8612	0.005694
8	**4**	225.4619	273.5044	148.1604	**100.0000**	94.4775	0.005591
9	**5**	91.7556	184.2423	312.8093	390.1137	**100.0000**	0.005026
10	**3**	387.8563	243.3958	**100.0000**	223.3919	153.1747	0.006029
11	**4**	335.9041	347.5702	33.9964	**100.0000**	112.0200	0.005270
12	**4**	422.1961	172.2312	390.2598	**100.0000**	3.3577	0.005574
13	**3**	193.3856	457.9956	**100.0000**	231.2246	212.1745	0.006480
14	**1**	**100.0000**	161.2359	392.3696	235.6786	17.8814	0.008226
15	**1**	**100.0000**	236.7430	76.3606	170.5623	303.6946	0.004760
16	**1**	**100.0000**	121.4248	458.7122	134.5308	382.7500	0.006068
17	**3**	143.7491	45.5567	**100.0000**	341.6816	273.2966	0.005620
18	**2**	322.2214	**100.0000**	339.5084	317.8934	472.5871	0.006695
19	**3**	354.6409	118.1153	**100.0000**	303.6520	225.0688	0.005756
20	**5**	330.9724	385.1428	175.1090	331.0048	**100.0000**	0.006781

Using the function *mark* (*function M* = *mark* (*A*, *B*, *C*, *D*, *E*, *Maxpoint*, *Sn*)) the points obtained (the score for a problem, topic or term) is converted into the six-point system of marks. The function has 7 input arguments: the first five arguments (*A*, *B*, *C*, *D* and *E*) are the lower limits of the marks: "Excellent 6", "Excellent", "Very Good", "Good" and "Satisfactory". These values are intended to be coordinated with the scale of the State matriculation examinations for the previous school year. For the school 2012 – 2013 year, they are as follows: $A = 95$; $B = 77$; $C = 59$; $D = 41$; $E = 23$ (Table 2). The sixth input argument in the function *mark* is *Maxpoint*, indicating the maximum number of points per problem / topic / term (as the maximum number of points is not fixed at 100, as in the State matriculation examinations) [3]. The seventh input argument is *Sn*, indicating the number of points accumulated by the student for a problem / topic / term, which will be transformed into a mark. The function has one output argument *M*, which is a string indicating the mark of the student.

In the function *mark* the interval which the points converted fall in is checked up. If the number of points received by the student at the moment *Sn* is greater than or equal to $A * Maxpoint / 100$, then the function takes the value 'Excellent 6'. If *Sn* is greater than or equal to *B* * *Maxpoint* / 100, then *M* = 'Excellent'. If *Sn* is greater than or equal to *C* * *Maxpoint* / 100, then *M* = 'Very Good'. If *Sn* is greater than or equal to *D* * *Maxpoint* / 100, then *M* = 'Good'. If *Sn* is greater than or equal to *E* * *Maxpoint* / 100, then *M* = 'Satisfactory', otherwise the mark is defined as *M* = 'Poor'.

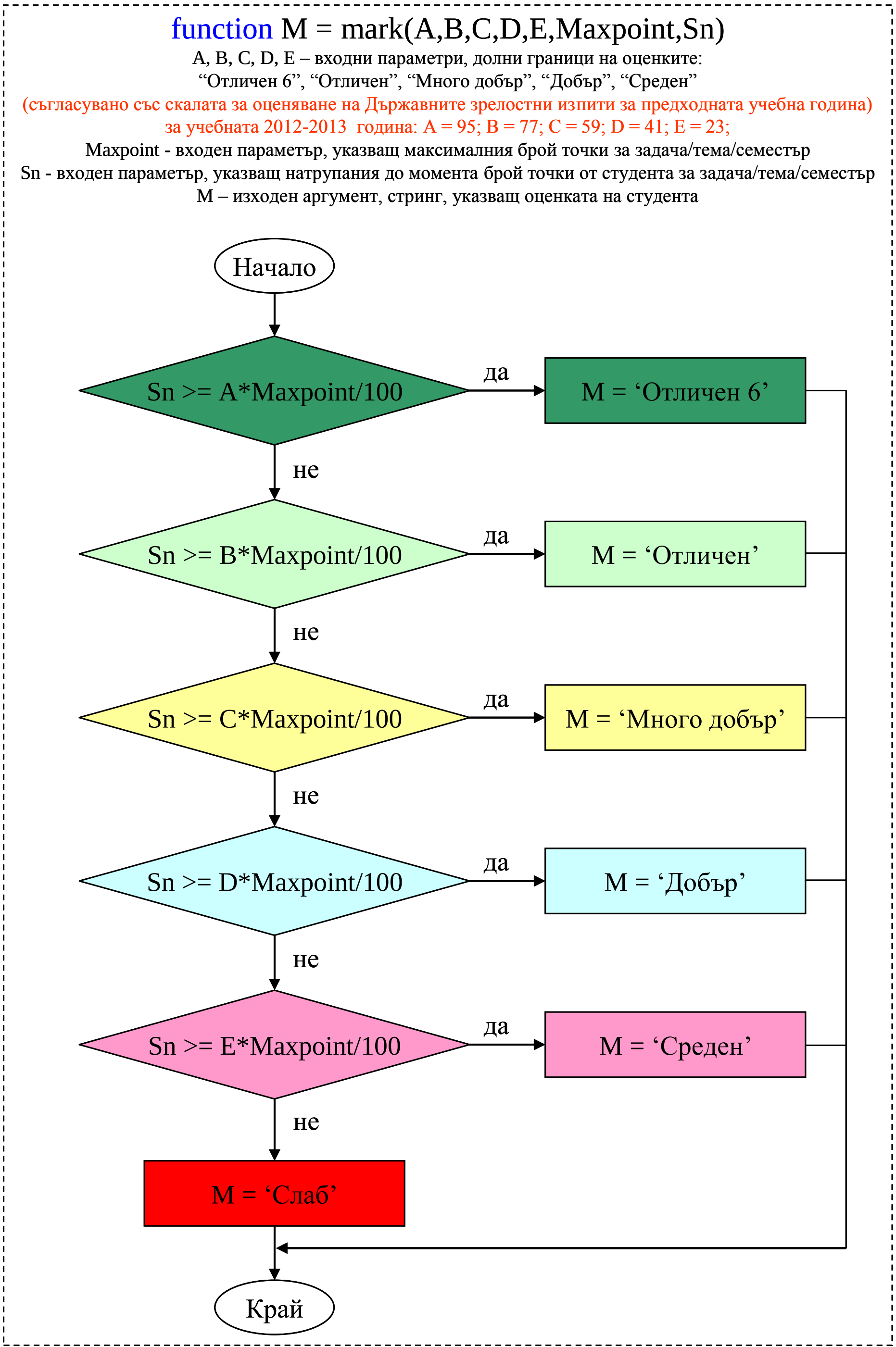

Fig. 2. Block diagram of the algorithm of the function *mark*
for converting the number of points into marks

At the moment in the function *mark* realized the mark formed by the test module is not specified to the nearest hundredth, but improvement of the function is expected.

3. TEST MODULE – SIMULATION RESULTS

The test module created shows the value for each quantity a number of possible values and the student must choose one of them. When the answer is right or wrong the student receives or loses the point / points. Then the test module forms the mark of the student for a problem / topic / term according to pre-set scale. The number of points is set by the teacher for each calculation, depending on its complexity.

Table 2. Scale for transforming the number of points in marks in the State matriculation examinations for the academic 2012-2013 year

State Matriculation Examinations - assessment scale

Scale for conversion of points in marks	
MARK	**POINTS**
Poor 2	up to 22,5 points incl.
Satisfactory (3,00 - 3,49)	from **23 to** 40,5 points incl.
Good (3,50 - 4,49)	from 41 to 58,5 points incl.
Very Good (4,50 - 5,49)	from 59 to 76,5 points incl.
Excellent (5,50 - 5,99)	from 77 to 94,5 points incl.
Excellent 6	from **95 to** 100 points

A further expansion of the test module to increase the types of problems to be solved during the practical exercises in the course "Communication Circuits" is expected.

In Fig. 3 the principle of operation of the test module in solving a problem in the topic "Amplitude modulation" (Problem 4.6.23, [1]) is demonstrated. The operation of the test module is given for two computed values (the index and the percentage of amplitude modulation) with the aim of visualizing the result on a single screen.

In block 1 of Fig. 3 the input data to solve the problem is entered: the power of the carrier ($P_c = 5\,W$) and the total power ($P_t = 5{,}625\,W$, when the second input numerical value is entered) as the correctness of the input data is checked. An improper input value of the total power ($P_t = 3\,W$) is indicated with an arrow because the value entered is less than the power of the carrier.

In block 2 and block 3 of Fig. 3 the principle of operation of the test module is illustrated: the script *MD*1.*m* calculates the values of the index and the percentage of amplitude modulation, then calls the function *test*, for generating five possible options for answers. The student must choose one of them – the value closest to the calculated one by him/her (if properly calculated).

For calculating the index of amplitude modulation in the formula $m_a = \sqrt{2(P_t / P_c - 1)}$ yielding the value $m_a = 0{,}5$, the teacher defines as weight of 3 points, i.e. receiving 3 points for a correct answer or taking off 3 points for a wrong answer. The function *test* displays a warning message. The percentage of amplitude modulation is obtained $m_a = 50\,\%$ and because of ease of calculation (by multiplying by a factor of 100), the teacher defines as weight of 1 point, i.e. receiving 1 point for the correct answer or taking off 1 point for the wrong answer, the function *test* displays a warning message also.

In block 2 of Fig. 3 it can be seen that the correct answer (0.5) is located at position 3. When selecting the third position by the student, i.e. the correct answer, the function *test* shows the message “You answered correctly! The number of your points has just increased and now they are 3.”.

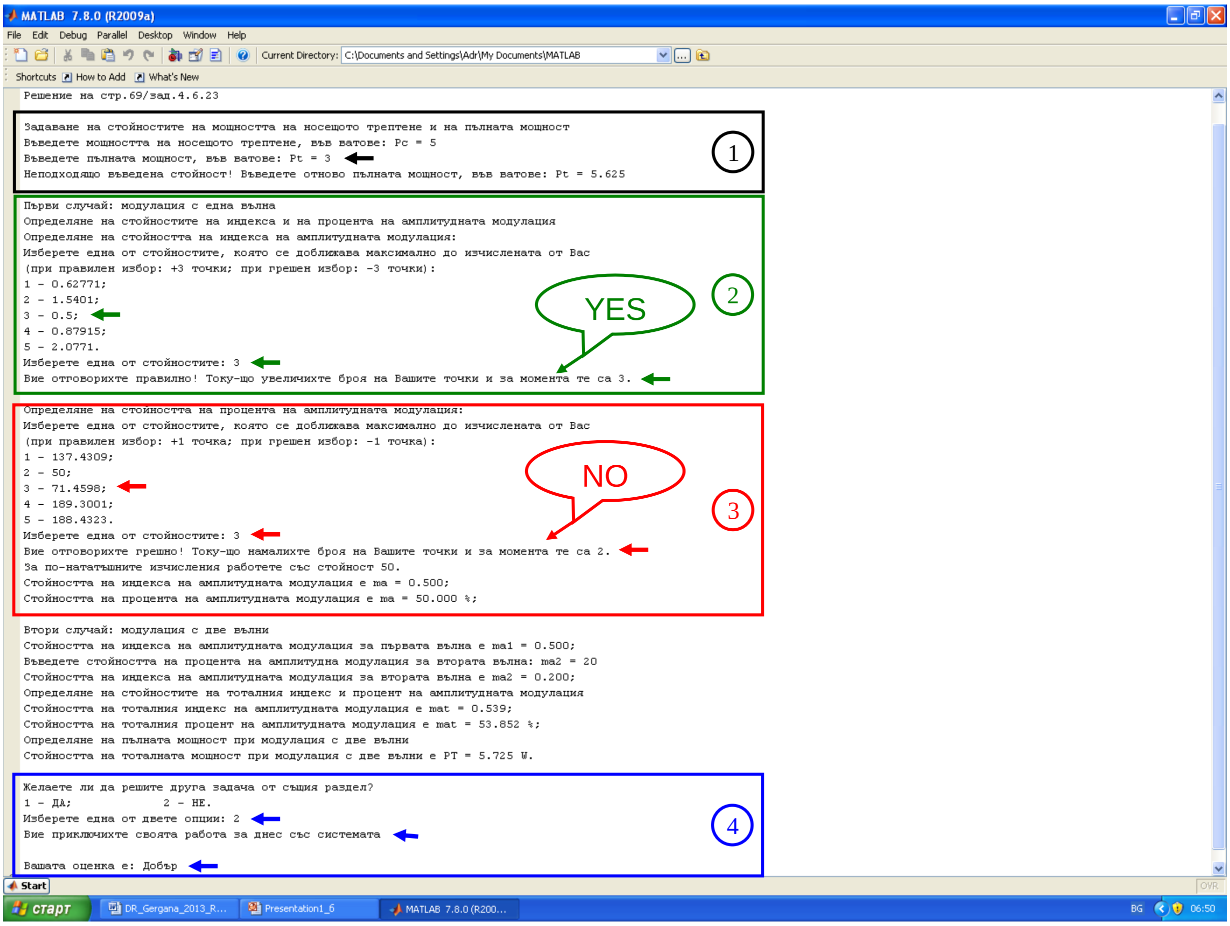

Fig. 3. Principle of operation of the test module in solving a problem
for amplitude modulation (Problem 4.6.23)

In block 3 of Fig. 3 it can be seen that the correct answer (50) is located at position 2 now. When selecting the third position by the student, i.e. one of the wrong answers, the function *test* shows the message “You answered wrong! The number of your points has just reduced and now they are 2.”. The correct value is displayed by the message: “For further calculations use the value 50”. It is done with the aim not to accumulate more errors in the process of further problem solving. Of course, the student has already been “punished” by taking off points for the wrong answer. Position of the correct answer is random (number from 1 to 5).

After the end of solving each problem (Fig. 3, block 4) a message: “Would you like to solve another problem in the same section?” with two possible answers: “Yes” (by pressing the button “1” on the keyboard) and “No” (by pushing the button “2” on the keyboard). If the student wishes to solve another problem in the same section a menu for choosing a topic / problem appears. If the student does not want to solve another problem, the system shows the message: "You finished your work for the day with the system". Then the final mark of the student is formed. This is realized by the function *mark*. The student can see that he has been assessed for both the answers with a mark "Good". It means that the student must be careful when entering values, respectively when calculating the desired quantities.

4. SPECIAL FEATURES OF THE SOFTWARE IMPLEMENTATION USING MATLAB

In the architecture of the software system developed the scripts for solving the problems in the individual topics (*TK*1.*m*, *TK*2.*m*, *TK*3.*m*, *TK*4.*m*, *EF*1.*m*, *EF*2.*m*, *EF*3.*m*, *EF*4.*m*, *MD*1.*m*, *MD*2.*m*, *MD*3.*m* and *MD*4.*m*) [4], respectively, were transformed into pseudo-code using the command *pcode* and are stored in files with the extension .*p* (*TK*1.*p*, *TK*2.*p*, *TK*3.*p*, *TK*4.*p*, *EF*1.*p*, *EF*2.*p*, *EF*3.*p*, *EF*4.*p*, *MD*1.*p*, *MD*2.*p*, *MD*3.*p* and *MD*4.*p*). When trying to open such a file (in this case *MD*1.*p*) a message that Windows can not open this file and the user must choose the kind of software to open it is obtained (Fig. 4). In Windows Explorer, both files are shown as files on MATLAB (MATLAB *m*-file and MATLAB *p*-code), but *p*-code can not be opened with MATLAB Editor (unlike file *MD*1.*m*, which will automatically be opened).

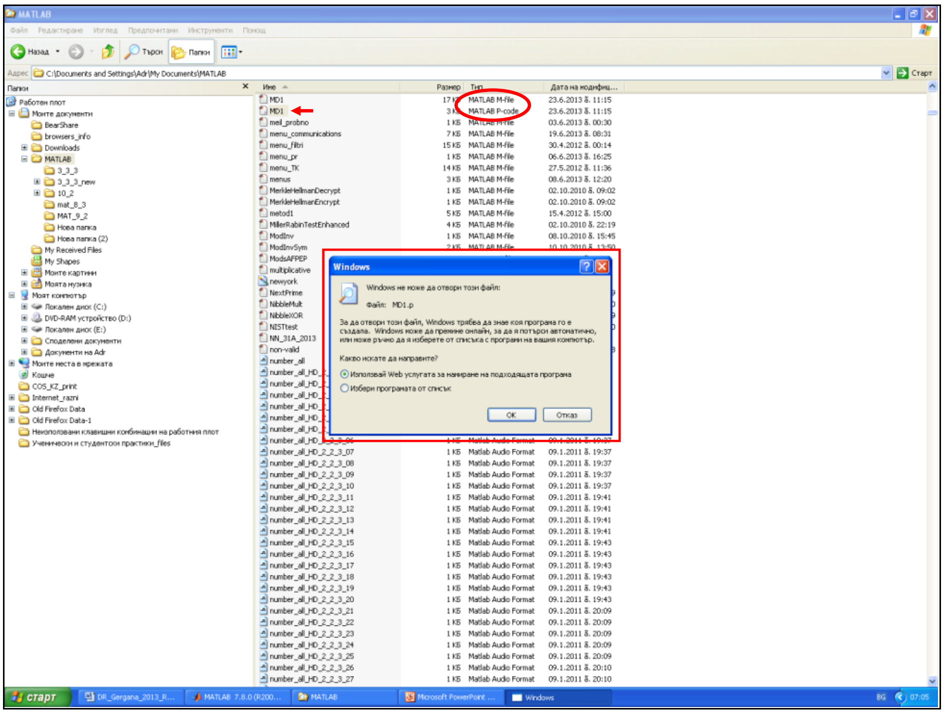

Fig. 4. Illustrating the impossibility of recognizing the file *MD*1.*p* (with extension .*p*), containing the pseudo-code of the script *MD*1.*m*

When choosing a widespread product Notepad, this file can be opened (Fig. 5), but the code of the program in it is hidden from the user (in this case students) – the file looks like an encrypted file.

The software system developed stores the student's work during the practical exercise in a text file. The way of forming the name of the text file is inspired by the method for creating e-mails to students at the University of Ruse, namely *sXXXXXX.txt*, where *XXXXXX* are the six digits of the faculty number of the student. It could also include a number indicating for example, the number of the exercise.

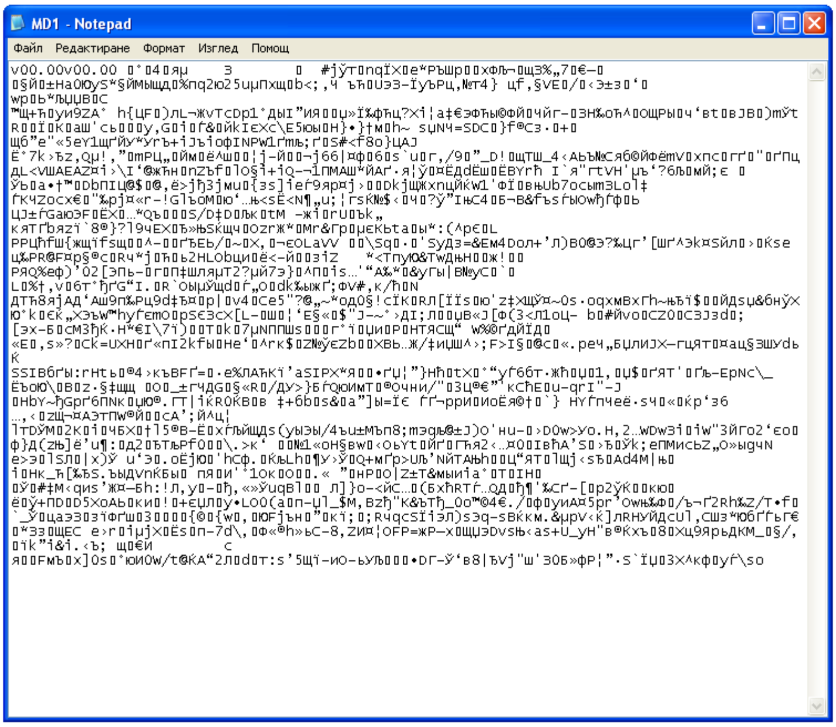

Fig. 5. Visualizing the file *MD*1.*p* (with extension .*p*), containing the pseudo-code of the script *MD*1.*m*

Using the command *diary sXXXXXX.txt* recording MATLAB-session in the text file *sXXXXXX.txt* is done, i.e. all subsequent MATLAB commands and results of their implementation (no graphics) are stored in the file with the given name. The command *diary* with another file name switch entries in the new file. Contents of the file *s*113303.*txt* (with extension .*txt*) with the data from the current MATLAB-session of the student with faculty number 113303 is visualized through Notepad (Fig. 6).

In block 1 of Fig. 6 all MATLAB commands and results of their execution in the current session of the student are displayed. In block 2 the data of the student, such as the academic year, the administrative group, the names and faculty number of the student, the section, the topic and the mark of the student, formed by the activity of the test module of the system is displayed.

5. CONCLUSIONS

The test module of a software system developed for synthesis and analysis of communication circuits and processes is presented in the paper. The test module is designed to objectively assess the knowledge and skills of students allowing the teacher to form the final mark for a problem / exercise / term (for each student). It will be used in the course “Communication Circuits” included as mandatory in the curriculum of the specialty “Telecommunication Systems” for the Bachelor degree at the University of Ruse.

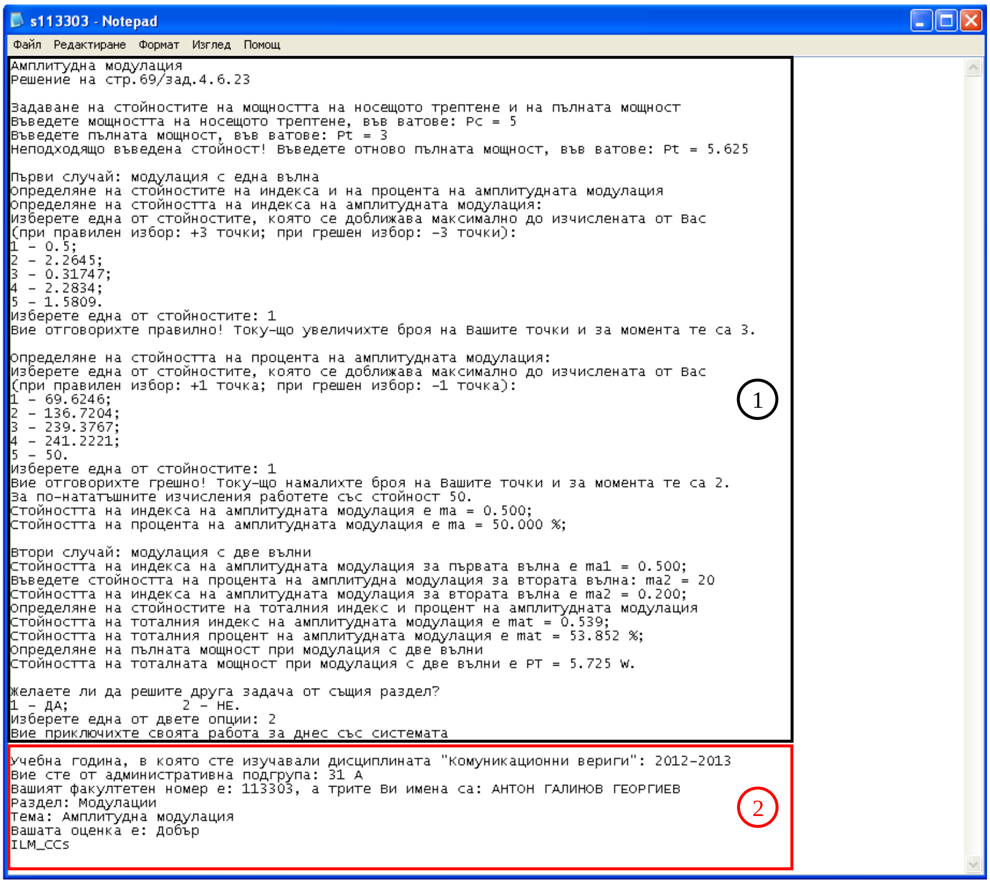

Fig. 6. Visualizing the contents of the file *s*113303.*txt* (with extension .*txt*), containing the data from the current session of the student with faculty number 113303 using Notepad

References

[1] Manukova, A., A. Borodzhieva. Communication circuits – a handbook for exercises. Ruse, University of Ruse, 2002 (in Bulgarian).

[2] http://ecet.ecs.uni-ruse.bg/else

[3] http://mon.bg/opencms/export/sites/mon/top_menu/general/dzi/dzi2008/08-06-13_dzi_skala_ocenjavane.pdf

[4] Borodzhieva, A., T. Stoyanova. Software System for Synthesis and Analysis of Communications Circuits and Processes Using MATLAB. 10th Summer School, Advanced Aspects of Theoretical Electrical Engineering, Sozopol '2014, 19.09. –22.09.2014, Sozopol, Bulgaria, (in the press).

Reviewer: Prof. PhD K. Brandisky