# Testing system "Archimedes"
# for local or remote assessment of student's knowledge

Borislav Slavov , Todor Dimov*

*Abstract: The testing system "Archimedes" is designed for assessment of student's knowledge in whatever discipline of university curriculum. The determination of the field's discipline is given by generating of suitable database, containing questions and answers for the curriculum material. The dialog between the system and the tested student is clear and precludes any difficulties (misunderstandings). The system use allows the tested studying material and the visual environment both to be introduced in Latin alphabet (English) or in Cyrillic alphabet (Bulgarian). The system can be downloaded through the Internet by the customers, who want to check the level of their knowledge in certain subject. The system is based on the University server and its downloading could be done from the department of Programming and.computer system applications – "PIIS". The tested student carries out the test by himself, using the computer system connected with Internet and via e-mail returns the result as file. The tested student doesn't have the possibility neither to change the content of the database, nor to check and alter the test results. The subject tutor checks the students tests with the system and publishes in the department site the test results. The system gives the opportunity of group test in computer rooms with flock of students in different disciplines. The test results can be transferred through local network in the respective folder "Test N" in the computer of the tutor who tests the students. Using the system by the customers doesn't require any special hardware and software resources. Fairly enough is presence of computer system with operating system Windows 95+.*
*Key words: assessment system, databases, tests, weight coefficients, C++*

### INTRODUCTION

The program is a testing system for assessment of student's knowledge in certain subjects. The information material is presented as questions and answers. The number of answers is unspecified and can be different for each question. Accepted answers are more than one true answer to certain question. Each question from the test is assessed trough marks (2) to (6) on the basis of indicated false and true answers by the tested student. Skipped question from the test (indicated no answer) is treated as poor mark. Indicated all answers to certain question (true and false) also is treated as poor mark Tree approaches are used to access each question of test (level of assessment 0,1, 2). In level 2 there is a possibility for designing a coefficient - "weight coefficient" - weight of the answer to the question. The final mark of the test is defined as average mark from the marks of the separate questions. The number of questions in the test is designed by the testing tutor, as the numbers of questions are chosen at random by the use of random generator. The test done by the student is saved as output file with title corresponding to his faculty number and extension "tst" (for example AU-156-2). Checking the student's test is carried out automatically by the system, as results are recorded in output file with title "Arhimed.txt". Access to this file is given only to the testing tutor.

## Description

The databases of the testing system "Archimedes" consists of the following files:

- **"Students.db"** - contains information about the students(names and faculty numbers)
- **"Questions.db"** - contains all questions included in the test;
- **"Answers.db"** - contains the answers to all questions of the test and their "weight coefficients".

For realization of the physical structure of the system and the database, the program language C++ is used as for work environment is chosen C++ Builder. The structure of the database is built up with objects included in the dynamic memory of the computer, as linear lists linked at one. The physical storage of the database is in output binary files. The program description of the database structure as well as the access methods and working with objects are described in "class database" of the file "database.cpp" from the project. Fragments of the structure of this file are shown bellow:

```
//----------------------QUESTIONS DATABASE STRUCTURE--------
struct q_list
{          //-------------------------data-------------------------------------
           unsigned int   ID;                // "Question ID      "
           char  Question[size_is];          // "Question text     "
           char  Picture[40];                // "Image filename  "
           //-------------------------data-------------------------------------
           struct q_list *next;
};
//----------------------ANSWERS DATABASE STRUCTURE----------
struct a_list
{          //-------------------------data-------------------------------------
           unsigned int   ID;           // "Answer ID    "
           unsigned int q_ID;           // "Question ID  "
           char  Answer[size_is];       // "Answer  text "
           float Points;                // "Points            "
           //-------------------------data-------------------------------------
           struct a_list *next;
};
//----------------------TESTS DATABASE STRUCTURE---------------
struct t_list
{          //-------------------------data-------------------------------------
           unsigned int   ID;           // "Test ID       "
           unsigned int q_ID;           // "Question ID"
           unsigned int a_ID;           // "Answer ID  "
           float Points;                // "Points            "
           //-------------------------data-------------------------------------
           struct t_list *next;
};
//----------------------STUDENTS DATABASE STRUCTURE--------
struct s_list
{          //-------------------------data-------------------------------------
           unsigned int   ID;           // "Student ID     "
           char  Student[size_is];      // "Student Names "
           char  FNumber[10];           // "Faculty Number"
           //-------------------------data-------------------------------------
           struct s_list *next;
};
```

where "**size_is**" is defined global size constant (maximal number of symbols in each question and answer) and is equal to 120.Maximal number of questions, that can be included in a given test is equal to 500.

## Assessment

For assessing the answers to separate questions included in the test the following approach was chosen:

**Task assessment formulation:**

Let's suggest that each question has **N –** number of answers, from which **At** are true and **Af** are false. The tested student can indicate **Bt** true answers to the question and **Bf** false answers to the question. It is necessary to put a mark of the question, which must be within the framework of poor (2) to excellent (6).

**Decision options:**

**I. Assessment without determining the weights of the answers in the question.**

The total amount of answers to the question is: N=At + Af

1. Introduce the following coefficients:

$Tt = \dfrac{At}{N}$   - coefficient of the total number of true answers;

$Tf = \dfrac{Af}{N}$   - coefficient of the total number of false answers;

$St = \dfrac{Bt}{N}$   - coefficient of indicated true answers given from tested student

$Sf = \dfrac{Bf}{N}$   - coefficient of indicated false answers given from tested student

2. Coefficients **T**t  and  **T**f  divide to 5 ( about marks: 2 , 3 , 4 , 5 , 6 )

$dTt = \dfrac{Tt}{5}$        and        $dTf = \dfrac{Tf}{5}$ ,

where the borders of each assessment framework are:

| | | |
|---|---|---|
| Tt1=1*dTt | and | Tf1=1*dTf |
| Tt2=2*dTt | and | Tf2=2*dTf |
| Tt3=3*dTt | and | Tf3=3*dTf |
| Tt4=4*dTt | and | Tf4=4*dTf |
| Tt5=5*dTt | and | Tf5=5*dTf |

3. Coefficient **St** is checked in which assessment framework will fall in, and depending on it, a concrete mark is given - **level 0**:

St = [ 0.0 ÷ Tt1 ]    - mark (2) is given
St = ( Tt1 ÷ Tt2 ]   - mark (3) is given
St = ( Tt2 ÷ Tt3 ]   - mark (4) is given
St = ( Tt3 ÷ Tt4 ]   - mark (5) is given
St = ( Tt4 ÷ Tt5 ]   - mark (6) is given

4. Coefficient **Sf** is defined in which framework will fall in and the given mark from point 3 (if the mark is different from poor 2) is filtered with coefficient **Rk** - **level 1**,

where:

$Rk = \dfrac{Bf}{Af}$   and corrected marks are formed according the following scheme:

Sf = 0                   -   final mark = mark from point 3;
Sf = ( 0.0 ÷ Tf1 ]   -   final mark = (mark - 1 x **Rk**);
Sf = ( Tf1 ÷ Tf2 ]   -   final mark = (mark - 2 x **Rk**);
Sf = ( Tf2 ÷ Tf3 ]   -   final mark = (mark - 3 x **Rk**);
Sf = ( Tf3 ÷ Tf4 ]   -   final mark = (mark - 4 x **Rk**);
Sf = ( Tf4 ÷ Tf5 ]   -   final mark = poor (2).

In addition:
⇒ if the mark from point 3 is poor(2) then the final mark is poor(2);
⇒ if calculated final mark is less or equal to 0 then the final mark becomes poor (2).

**II. Assessment with determining the weights of the answers in the question.**
All answers in certain question form two multitudes:

-multitude of true answers [**BO**]:
  $BO_1$ , $BO_2$, $BO_3$, ... , $BO_к$
 with weights:
  $U_1$ , $U_2$ , $U_3$ , ... , $U_к$
 - multitude of false answers [**HO**]:
 $HO_1$ , $HO_2$, $HO_3$, ... , $HO_p$
 with weights:
 $W_1$ , $W_2$ , $W_3$ , ... , $W_p$ ,
  where total number of answers is: N=(к+p).
The student's answers form two sub multitudes:
  - sub multitude of true answers [**CBO**]:
   $CBO_m$ , $CBO_q$, $CBO_s$, ... , $CBO_t$
  with weights:
  $U_m$ , $U_q$ , $U_s$ , ... , $U_t$
   where indexes (m, q, s, …, t) are within the framework{ 1 ÷ к }
  - sub multitude of false answers [**CHO**]
   $CHO_f$ , $CHO_g$, $CHO_h$, ... , $CHO_j$
   with weights:
   $W_f$ , $W_g$ , $W_h$ , ... , $W_j$ ,
    where indexes (f , g , h, …, j) are within the framework { 1 ÷ p }

1. Introduce the following coefficients:

$S_{total}= U_1 + U_2 + U_3 + …+ U_к$     - sum of weights of all true answers;
$S_{true}= U_m + U_q + U_s + …+ U_t$     - sum of weights of indicated true answers;
$S_{false}= W_f + W_g + W_h + ... + W_j$     - sum of weights of indicated false answers;
$S_{result}= S_{true} + S_{false}$       - resulting weight of indicated answers

$W_f<=0$; $W_g <=0$; $W_h <=0$; … $W_j<=0$;

2. Forming the assessment areas of the weights of all true answers.

$$dS = \frac{Stotal}{5} \quad ,$$

(-∞ ÷ dS ) - mark (2) is given
[dS ÷ 2dS) - mark (3) is given
[2dS ÷ 3dS) - mark (4) is given
[3dS ÷ 4dS) - mark (5) is given
[4dS ÷ 5dS] - mark (6) is given

3. The coefficient $S_{result}$ is checked in which areas it falls in and the mark is determined.
   **Interface**
      The testing system "Archimedes" has the following appearance:
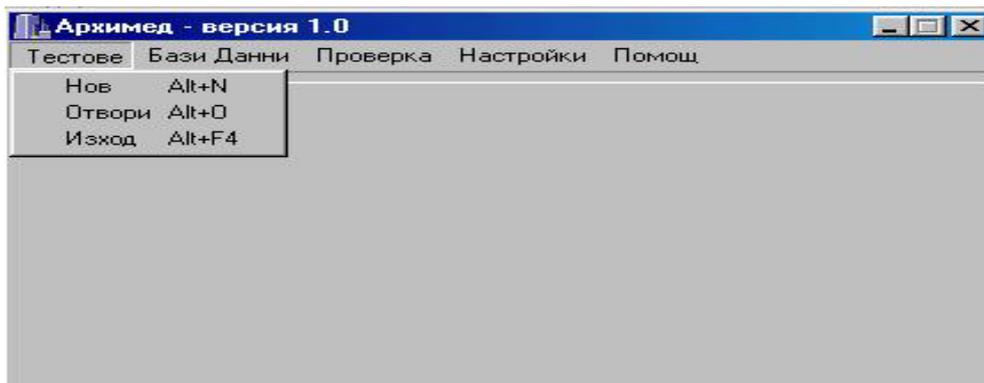
Form of the test



Student's registering form



Question's registering form

Answer's registering form
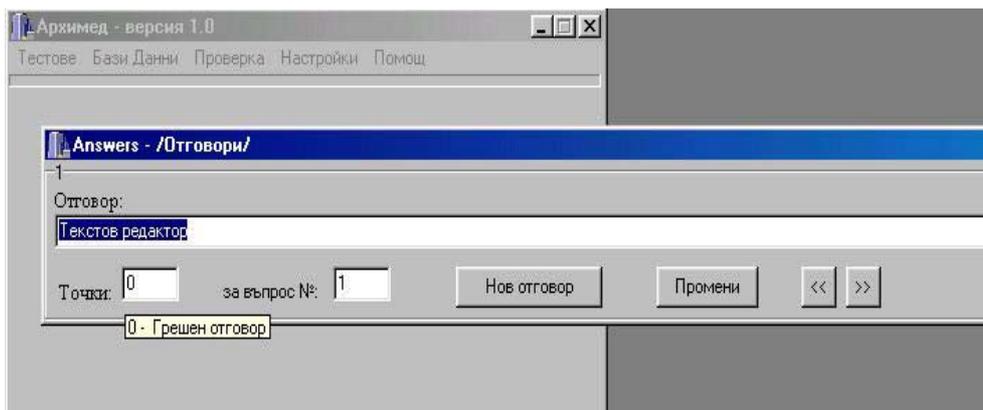


Testing system access changing form
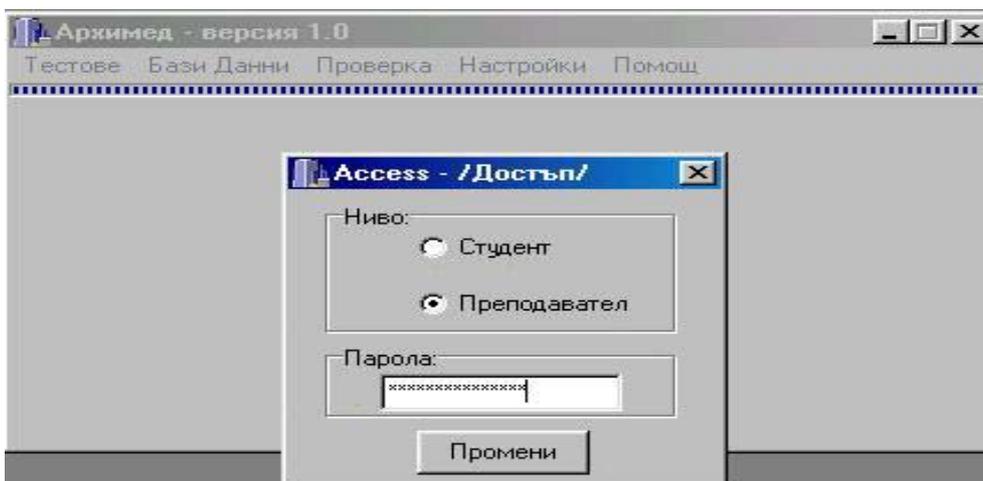


### CONCLUSIONS AND FUTURE WORK

The program system "Archimedes" is implemented successfully in the examination of a student flock of 64 students, part time education in Informatics I  - February 2004. The system is used by students to check the acquired knowledge in their self  preparation as well as by computer rooms at free access in the university and also by personal computer systems. Impending is the implementation of the system in the examination of bachelor students full time education (400 persons) in the discipline Informatics II - during the summer semester of 2003/2004.

### REFERENCES

[1]  Gradinarova B., Introduction of Education Software on Internet, IHPM, Varna 2001.

[2]  Klinachev N. V., Theory of the System for Automatic Adjustment and Control: Education and Methodical Complex., Cheliabinsk, 2003. (in Russian).

[3]  Minchev V., Dimov T. Contemporary means for the development of a programmed problem, Journal of UCTM-Sofia, Vol.XXXVIII, 2003.

[4]  Raven J. The tragic Illusion: Educational Testing, Oxford, 1999, p. 141

### ABOUT THE AUTHORS

Assoc. Prof. Todor Dimov, PhD, Department of Computer Systems Applications, University of Chemical Technology and Metallurgy, Sofia, Phone: +359 02 6254 (624), e-mail: tdimov@uctm.edu.

Ass. Prof. Borislav Slavov, Department of Computer Systems Applications, University of Chemical Technology and Metallurgy, Sofia, Phone: +359 02 6254 (626), e-mail: bslavov@uctm.edu.