2. Krizhanovskaya N.Y., Gordienko Y.S., Degtev I.A. Methods of forming prirodointegrirovannoy architecture in the urban environment: monograph // Belgorod: Publishing House of the Belarusian State Technological University, 2010 – 144 p.

3. Tabunschikov Y.U., Brodach M.M., Shilkin N.V. Energy efficient buildings. – Moscow: Avrora Press, 2003. 200 p.

4. International Forum "Strategy of development of housing construction in Russia" [electronic resource]. Requirements: AdobeAcrobatReader: http: //www.stroy-conference.ru.

5. Smirnov S.N. The oretical model of energy-efficient residential building / Scientific journal. Periodic scientific publication. – Nizhny Novgorod: NNGASU, 2009. – P. 86 – 91.

6. Chernysh N.D., Tarasenko V.N. The microclimate of residential areas as a multicomponent medium architectural design // Belgorod: Bulletin of BSTU named after V.G. Shukhov. 2015. № 6. p. 57 – 61.

7. Savchenko E.S., Gridchin A.M., Lesovik V.S., Smolyago G.A. Conceptual approaches to solve the housing problem in the Russian Federation on an example of the Belgorod Region: Virtual exhibition of energy saving. RJ 20T. Construction Economics. 2006. № 11. p. 83.

8. Tarasenko V.N. Design of Noise Protection Structures // High technology and innovation: international scientific-practical conference dedicated to the 60-th anniversary of Belgorod state of BSTU named after V.G. Shukhov (XXI scientific readings). Belgorod: Publishing house of BSTU, pp. 115 – 117.

9. Tarasenko V.N., Soloveva L.N. The Problem of Sound Insulation in Housing Construction . Belgorod: Bulletin of BSTU named after V.G. Shukhov. 2013. № 4. p. 48 – 52.

10. Lesovik R.V., Botsman L.N., Tarasenko V.N. Enhancement of Sound Insulation of Lightweight Concrete Based on Nanostructured Granular Aggregate // ARPN Journal of Engineering and Applied Sciences. 2014. № 10, p. 1789 – 1793.

11. Tarasenko V.N., Degtev I.A., Golikov G.G. Study of the Noise in the Hall Multi-Purpose SDK Students BSTU V. G. Shukhov // Belgorod: Bulletin of BSTU named after V.G. Shukhov. 2016. № 5. p. 39 – 44.

12. Tarasenko V.N., Degtev I.A., Chernysh N.D. Acoustic Comfort Multi-Purpose Hall of the Palace of Culture Students of the BSTU V.G. Shukhov // Belgorod: Bulletin of BSTU named after V.G. Shukhov. 2016. № 6. p. 29 – 34.

13. Lesovik V.S. Building Materials. The Present and the Future // Moscow: Bulletin of MGSU. 2017. № 1 (100). p. 9 – 16.

14. Zagorodnjuk L.H., Lesovik V.S., Volodchenko A.A., Yerofeyev V.T. Optimization of Mixing Process for Heat-Insulating Mixtures in a Spiral Blade Mixer // International Journal of Pharmacy and Technology. 2016. № 3. p. 15146 – 15155.

15. Volodchenko A.A., Lesovik V.S., Zagorodnjuk L.H., Volodchenko A.N., Aleksandrovna K.A. The Control of Building Composite Structure Formation Through the Use of Multifunctional Modifiers // Research Journal of Applied Sciences. 2016. № 12. p. 931 – 936.

16. Morozova M.M., Frolova M.A., Lesovik V.S., Aizenshtadt A.M., Aizenshtadt A.M. Sorption Properties of Mineral Modifier for Frost Resistant Concrete // International Multidisciplinary Scientific GeoConference SGEM. 2016. № 1. p. 305 – 312.

Mazorchuk Mariia
PhD, Assistant Professor of Informatics Department,
National Aerospace University "Kharkiv Aviation Institute"
Dobriak Viktoriia
PhD, Assistant Professor of Informatics Department,
National Aerospace University "Kharkiv Aviation Institute"
Chumachenko Dmytro
Teaching Assistant of Informatics Department,
National Aerospace University "Kharkiv Aviation Institute"

ADAPTIVE TESTING TECHNOLOGY IN R

Summary. Currently much attention is paid to distance learning. The problem of design and development of adaptive testing systems that are effective not only in assessing the level of training, but in organizing a flexible process of distance learning based on the student's individual abilities is relevant. Existing systems of computer adaptive testing are quite expensive. Open-source software environment R allows implementing a number of features of adaptive testing and *mirtCat* package based on functions of such packages as *shiny* and *mirt*, allows realizing the mechanism of input-output test items through the web interface, and doing a test analysis possible on the basis of one-dimensional and multidimensional IRT models theory.

Keywords*:* distance learning, massive open online courses, computer adaptive testing, test items, test quality

Statement of the problem and analysis of recent research and publications. Currently, in connection with the development of massive open online courses and distance learning systems, computer-based testing technology is used widely [1-2]. This caused by that computer testing is effective, and often is the only possible way to control the distance learning process. Methods and approaches of computer adaptive testing (CAT) are paid a lot of attention [3-5]. Basically CAT methods are based on the models of modern IRT testing theory [6-8], based on the calculation of the model parameters by maximum likelihood.

In works [9-12] usage of spline models, which allows obtaining the characteristics of the test tasks in an automated mode with higher accuracy, is proposed for the implementation of the CAT. However, despite the rapid pace of development of methods and approaches in pedagogical testing, the use of methods of computer adaptive testing is still limited. This is due to several factors:

– difficulty in understanding and implementation of computer adaptive testing techniques based on models of modern IRT testing theory;

– need to create surround bank of calibrated test items;

– absence of effective methods of adaptive testing for a variety of academic disciplines and different test populations.

This is not the whole list of problems that lead to difficulties in the use of adaptive tests at computer evaluation. Also, there are problems of identification of students passing the test. There are problems associated with the contextual characteristics of students (gender, age, country of residence, education level, etc.), which lead to low results of passing the tests and the preschedule termination of training exist in the massive open online courses. The basic problem with CAT using is the complexity of the mathematical and algorithmic models and methods forming the basis of operation of the adaptive test, which, in turn, makes it impossible to implement a CAT process to conventional instructor, for example, to their distance courses. For example, some of the problems associated with the use of models in the CAT process are presented in [13-16].

Therefore the rationale for the selection of tools, as well as the development of methods of computer adaptive testing that will allow quick and efficient realization of the CAT process on the basis of modern computer equipment is relevant.

The **objective** of this paper is to analyze the modern tools that allow implementing the mechanism of adaptive testing, and development of methods of launching CAT-based HTML-interface on the example of tests on computer science.

Materials and Methods. General algorithm of CAT is shown in Figure 1. The process of CAT can be realized only on the basis of the calibrated bank of test items (BTI) and can be divided into 4 main stages. The first stage is the initial and involves selecting of one or more appropriate test items as the first item in the testing process. Medium difficulty items are selected usually. The second stage is the direct testing, which consists of the fact that items are sequentially selected from a bank of test items and the level of ability of the test-subject are re-assessed after each response. At this stage, if the subject does not respond to the question, he is given an easier task, and if he answers, the given task is more complicated. These steps are repeated until the stop test criterion is reached. The rules of termination of testing are determined at the stopping test stage. The final stage provides the final assessment of the abilities of a test-subject, and possibly other information about the student.

Many IRT models are used in CAT process. In this article we focus on the multidimensional four-parameter logistic models (M4PL) for dichotomous item (0 – incorrect answer, 1 – correct answer). The probability $P_j(y=1|\theta)$ that a examinee positively answer the j-th dichotomous item ($y=1$) with an M4PL [15, 16] structure is

$$P_j(y=1|\theta)=P_j\left(y=1\middle|\theta,a_j,d_j,g_j,u_j\right)=g_j+\frac{u_j-g_j}{1+e^{\left(-\left(a_j^{\mathrm{T}}\theta+d_j\right)\right)}}\quad , \tag{1}$$

where θ - D-dimensional vector of random ability or latent trait values; a_j – vector, which determine discrimination parameter of test item (this parameter influences the probability function); d_j - difficulty parameter of test item; g_j and u_j - parameters are restricted to be between 0 and 1 and determine parameters of guessing and inattentiveness respectively.

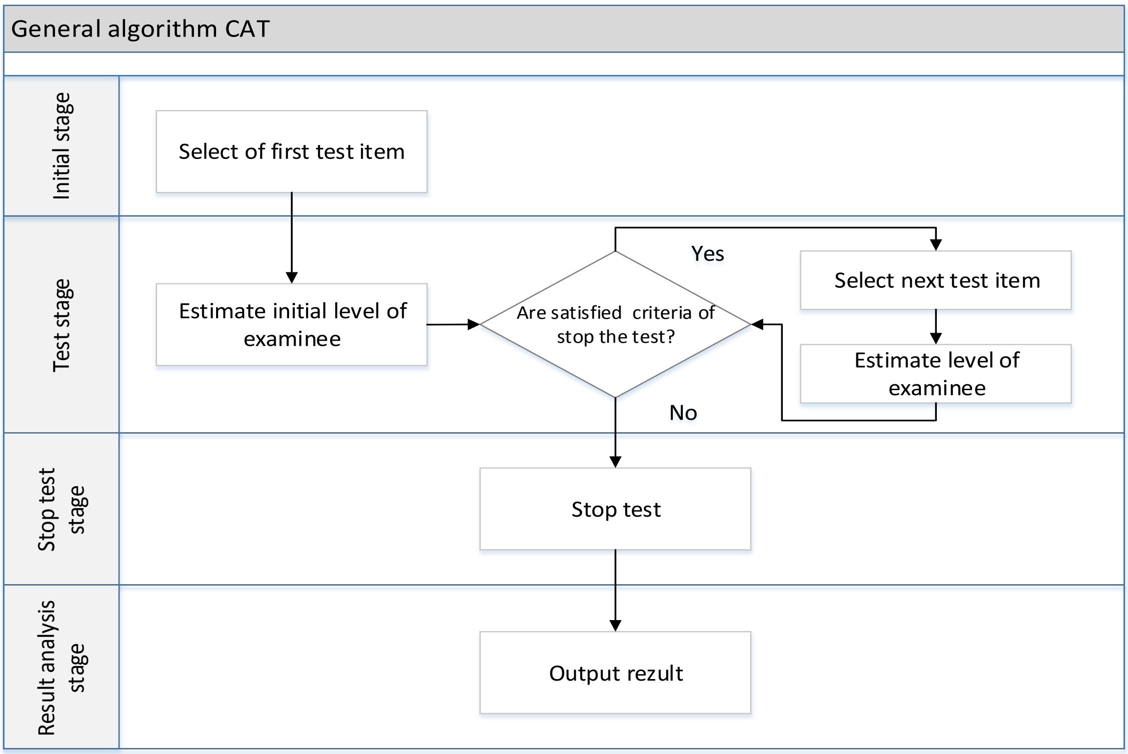

Figure. 1. Algorithm of CAT process

There are two groups among the existing systems which allow implementing adaptive testing can be distinguished: commercial and free. Commercial includes CATSim (Assessment Systems Corporation, 2012 [17]), Adaptest [18], Assessment Center [19], Winsteps [20] and others. The mechanisms implemented in these data packets allow simulating the process of adaptive testing, creating a bank of tests, calibrated according to IRT models and getting full statistics based on the results of testing. However, use of these packages is limited by high price. Free tools are developing front-end using C +, C #, Java, and others development platforms, and back-end using functions of R software-tool environment, which is open source environment for the processing of statistical data [21]. These systems include Firestar [22], CatIrt [23], CatR [24]. Among the latest non-commercial products, allowing realizing the mechanism of adaptive testing using a web interface, is a mirtCat package [25].

mirtCat package is based on the functions of such packages as the *shiny* package (which allows realization of the mechanism of input-output via a web interface) and *mirt* (dichotomous and multinomial test tasks analysis package based on one-dimensional and multidimensional models of IRT theory).

The main function of the package is a function *mirtCAT*, which, in essence, provides a tool for creating HTML-interface for the realization of adaptive test using *shiny* package. Function parameters define the methods of data processing according to the CAT algorithm (Figure 1). *mirtCAT* function has the following format:

```
mirtCAT(df = NULL, mo = NULL, method = "MAP", criteria = "seq", start_item = 1, local_pattern = NULL, design_elements = FALSE, cl = NULL, progress = FALSE, primeCluster = TRUE, design = list(), shinyGUI = list(), preCAT = list(), ...)
```

To output the adaptive test results, to display summary statistics of passing the test and to build the graph of test tasks selection by the test-subject functions *print(x, ...), summary(object, sort = TRUE, ...)* and *plot(x, pick_theta = NULL, true_thetas = TRUE, ...)* are used.

Full description function and function arguments of the *mirtCat* and *shiny* package can be find in [25-27].

Results and Discussion. With the use of the package the test of computer science for students of 1-st course of the National Aerospace University "Kharkiv Aviation Institute", allowing carrying out adaptive testing has been developed.

At the first stage, additional packages containing a set of functions that extend the basic features of the language R has been uploaded:

```
library('mirtCAT')
library('data.table')
```

In order to record data in *data.frame* each symbol value in the table is perceived as a value, not a factor, *false* value must be assigned to a logical parameter *stringsAsFactors*:

```
options(stringsAsFactors = FALSE)
```

The next stage is to fix the initial value for generating random numbers.

```
set.seed(1234)
```

In the next stage the data is read. Function *data.table :: fread ()* reads data from a txt-file in the specified path to the file with the test and creates a data table out of it (*data.frame*). Argument value *sep = "/"* assumes that the values of variables in a readable file are separated by *"/"*. The parameter value *header = TRUE* allows recording the first row from the read file to the column headers. Data is stored as a class object that allows realizing object-oriented access to data:

```
x<-data.table::fread
("E:/RstudioProject/Diplom/test.txt",sep="/",
header=TRUE)
    x <- as.data.frame(x)
    class(x)
```

We used another format file also. For example, if data is saved in Excel format, we can use *read_excel* from library *xlxs*:

```
    library(xlsx)
    x <- read_excel
("E:/RstudioProject/Diplom/test.xlsx", col_types =
c("text", "text", "text", "text", "text"))
    x <- as.data.frame(x)
    class(x)
```

Then the number of test questions is given using *nrow()* method:

```
    nitems <- nrow(x)
```

Answers patterns are simulated according to IRT models. First, the column headings are defined in the table with modeled answers: table header name - '*Item*' + number of the question in the test. Then, matrix *a* with the test tasks parameters (this matrix is responsible for such parameter of IRT model as the differentiating ability) is generated. For generating the lognormal distribution function (*rlnorm*) is used. *Nitems* parameters are the number of observations (in this case 10), 0.2 and 0.3 are the parameters for the law (mean and standard deviation of the distribution on a logarithmic scale with values).

```
    itemnames <- paste0('Item.', 1:nitems)
    a <- matrix(rlnorm(nitems, .2, .3))
```

In the next step we specify the initial matrix *d* coefficients (this matrix is responsible for such parameter of IRT model, as the complexity (difficulty) of test task). In the ideal case, the values of the matrix are equal to 0.5, which means that the candidate with a certain level of knowledge is able to answer given question correctly with 50% probability). For given matrix we use the function of answers generating based on the normal distribution (*rnorm*), where *nitems* is the number of questions, the mean and standard deviation of the distribution are 0 and 1, respectively (set by default).

```
    d <- matrix(rnorm(nitems))
```

Using *simdata()* function we simulate the answers patterns for MIRT models based on the matrixes a and d, where N is sample size, *itemtype* is a method of selecting the initial values in the new model. For the optimal calculation speed we choose *'dich'* method.

```
    dat <- simdata(a, d, N = 501, itemtype = 'dich')
```

Lets create the model of one-factor analysis (a_j are parameters reflecting the discrimination of items, d_j are parameters reflecting the complexity of items, $\mathrm{u_j} = 1$, $\mathrm{g_j} = 0$ are parameters of guessing and carelessness of IRT model). General view of the model is given by

$$P(\mathrm{X_j} = 1|\theta) = g_j + \frac{(\mathrm{u_j} - g_\mathrm{j})}{1 + e^{\left(-(a_j\theta + d_j)\right)}}$$

Function *mirt* allows generating patterns of answers, according to the two- or one-factor model, where *data* is pattern of answers for MIRT model, *model* is the variance of the latent factors. In our case, we use one-factor IRT model, therefore *model* is equal to 1, *itemtype* is type of IRT logistic model (in this case 2PL). Details about *mirt* package can be found in [27]. Calculation of the factor analysis coefficients for 2PL model is made by formula (2).

```
    mod <- mirt(data = dat, model = 1, itemtype =
'2PL')
```

Lets extract the indicative coefficients from model *mod*, using the *coef* function, where *simplify* is logical parameter indicating how to extract the coefficients (in this case with simplification):

```
    coef(mod, simplify=TRUE)
```

Lets create vectors from the data table, in which the test have been recorded: *Question* is vector with questions, *Option* is vector with possible answers for each question, *Answer* is vector with the correct answers for each question:

```
questions <- answers <- character(nitems)
options <- matrix("", nitems, 4)
spacing <- floor(d - min(d)) + 1

for (i in 1:nitems) {
 ans <- x[i, 2]
 questions[i] <- paste0(x[i, 1])
 answers[i] <- as.character(ans)
 v<-sample(1:4, 4, replace = F)
 ch <- sample(c(-4:-1, 1:4) * spacing[i, ], 4)
 ch[v[1]] <- ans
 ch[v[2]] <- x[i, 3]
 ch[v[3]] <- x[i, 4]
 ch[v[4]] <- x[i, 5]
 options[i, ] <- as.character(ch)
}
```

Lets specify options for *shinyGUI* function, which creates a visual shell of the adaptive test.

title <- " Multidimensional Computerized Adaptive Testing"
authors <- " Informatics department. National Aerospace University KhAI."
instructions<- c("Instructions:", " Click ‘Next’ to display the next page", "Next")
firstpage <- list(h2("Informatics test"), h5("Answer the following questions.
Test results will remain secured and it will be used for scientific purposes only."))
begin_message <- " Click next to start the test"
lastpage <- function(person)
return(list(h5("Your have completed your test! Click next to save the results.")))
shinyGUI_list <- list(title = title, authors = authors, instructions=instructions, firstpage = firstpage, begin_message = begin_message, lastpage = lastpage, stopApp = TRUE)
df <- data.frame(Question = questions, Option = options, Answer = answers, Type = "radio")

Lets define values for the *design* argument, which contains a list of parameters to control the adaptive test, where *delta_thetas* is stopping criterion of the adaptive test when factor θ is changed, *min_items* is the minimum number of questions that must be answered to complete the test:

design <- list(delta_thetas = 0.095, min_items = 2)

At the last stage we simulate multidimensional adaptive test with parameters: *df* is *data.frame*, containing vectors with data for the implementation of testing (questions, answers), *mo* is model of factorial analysis of most likelihood data, *method* is test parameters update criterion at the time of passing the test (final value θ is calculated using this method, in this case, using ‘EAP’ (Expected A Posteriori estimation function of ability level), *criteria* is sequence of questions supply in the multidimensional adaptive tests selection criteria (in this case, the selection criteria is *'Trule'* which is focused on trace information in the information matrix), *shinyGUI* is visual shell of the adaptive test.

result <- mirtCAT(df = df, mo = mod, method = 'EAP', criteria = 'Trule', shinyGUI = shinyGUI_list, design = design)

The output of result for multidimensional adaptive test:

print(result)

Output of graph, by which you can track the changes of θ, depending on answers to the test items (coordinate *x* is number of task, coordinate *y* is the range of values of θ).

plot(result)

User interfaces that reflect the stages of work with the test of computer science are shown in Figures 1 - 5.

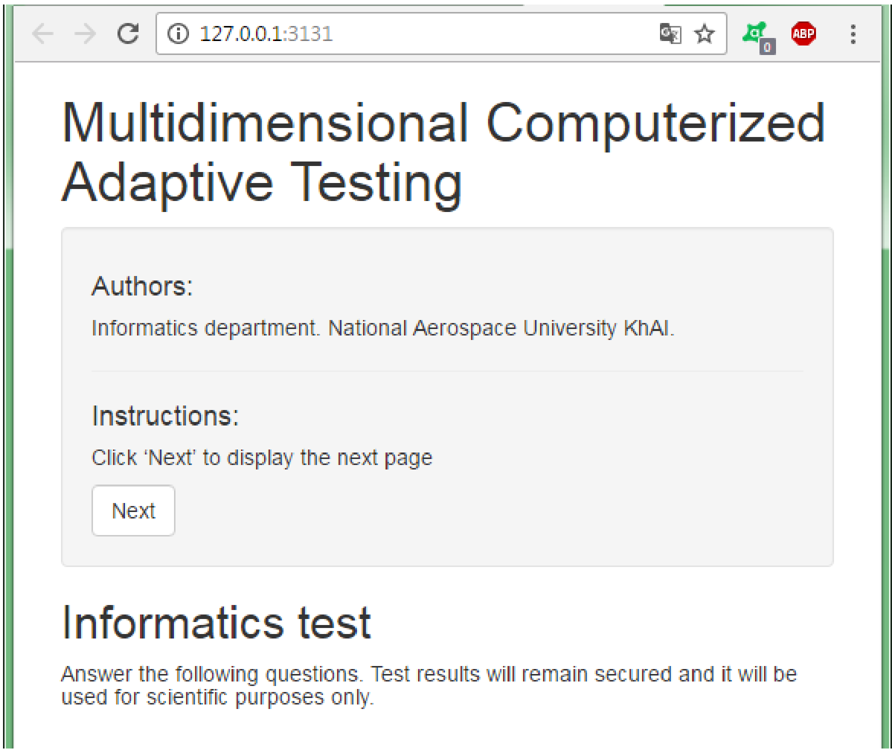

Figure 1 – First page with the adaptive test description

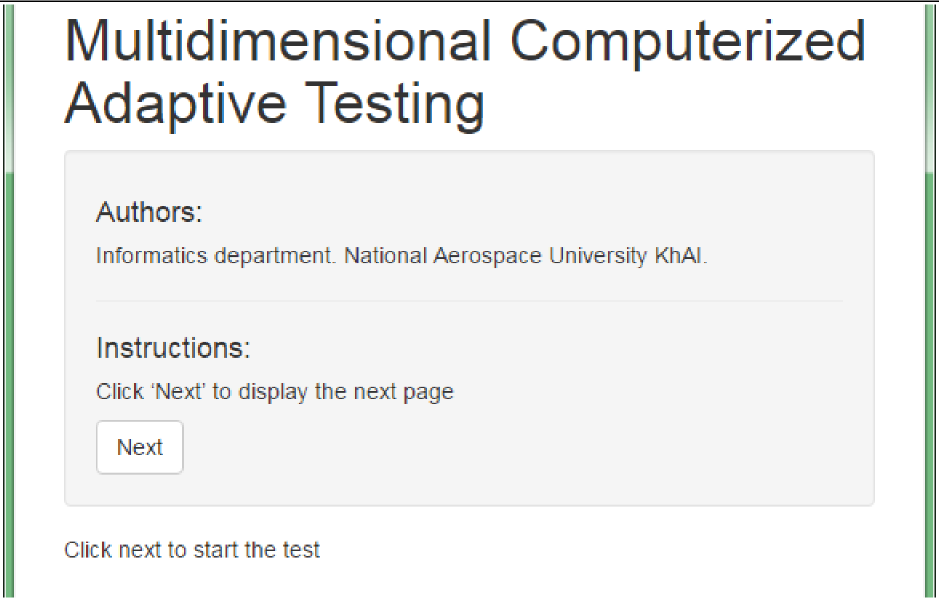

Figure 2 – Page of test instructions

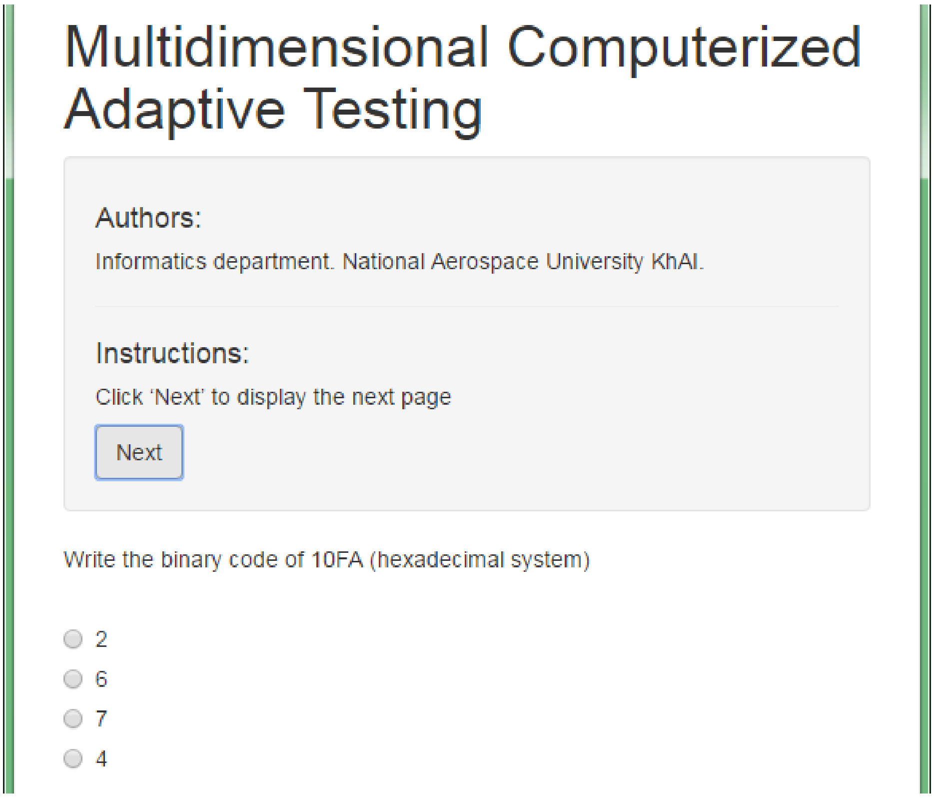

Figure 3 – Page with one of the questions of adaptive test

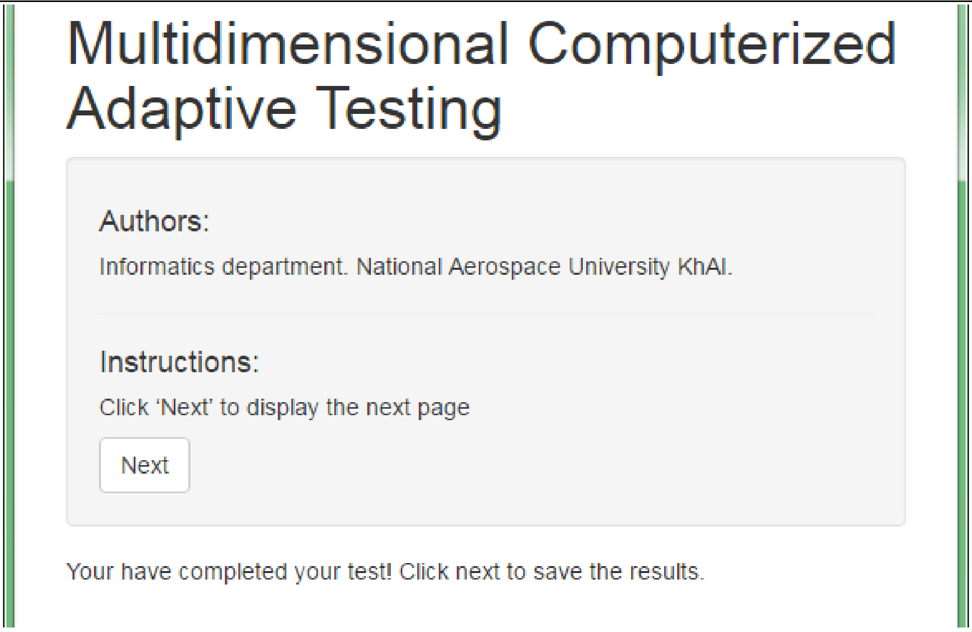

Figure 4 – Last page, that confirm that the adaptive test is finished

Then the test saving is carried out and its results can be viewed in R environment. Output of the results for the multidimensional adaptive testing is shown in Figures 5 - 8.

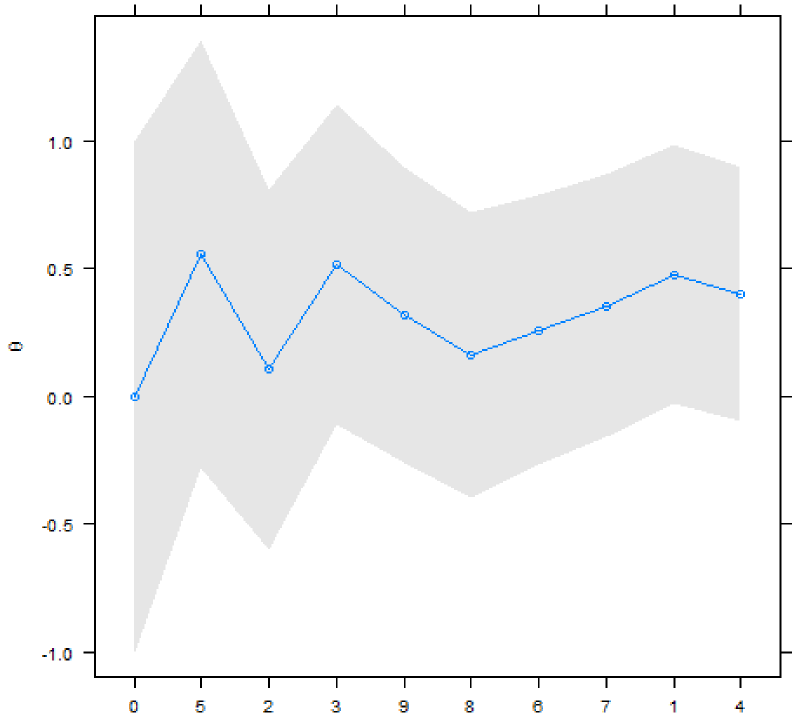

Figure 5 – Plot of change θ depending answers to test items

```
$scored_responses
[1] 1 0 1 0 0 1 1 1 0

$items_answered
[1] 5 2 3 9 8 6 7 1 4
```

Figure 6

The sequence of answers to test items and number of items in an adaptive test (0 – incorrect, 1 – correct)

```
$thetas_history
        Theta_1
 [1,] 0.0000000
 [2,] 0.5559839
 [3,] 0.1064812
 [4,] 0.5155796
 [5,] 0.3171088
 [6,] 0.1626613
 [7,] 0.2585814
 [8,] 0.3550165
 [9,] 0.4771229
[10,] 0.3998618
```

Figure 7 – Updates of parameters θ at the time of passing the test

```
 n.items.answered   Theta_1 SE.Theta_1
                9 0.3998618  0.4974189
$final_estimates
            Theta_1
Estimates 0.3998618
SEs       0.4974189
```

Figure 8 – The result of the adaptive test (values θ are the levels of preparation of the test-subject (Theta_1) and the average error (SE.Theta_1))

Conclusions. Thus, the above-mentioned method of computer adaptive tested in R environment allows developing an adaptive test (with tests templates, calibrated according to the multidimensional IRT model) and realizing it through the HTML-interface. Test development technology is not completely transparent, since used function of *mirtCAT*, *shiny*, *mirt* and other packages requires a certain understanding of the CAT and models process that are used to set criteria for the start of test, the testing process and test stopping. However, flexible functionality allows realizing CAT process with almost any initial conditions. Using *mirtCAT* package functions allow realizing the functional part of CAT systems developed in other programming environments.

References

1. Ruth C. Clark, (2011). E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning, 3 Edition, Pfeiffer.
2. Jordan K. (2015). Massive Open Online Course Completion Rates Revisited: Assessment, Length and Attrition. Available at: http://www.irrodl.org/index.php/irrodl/article/view/2112/3340.
3. Ying C., Computerized adaptive testing, New developments and applications. University of Illinois at Urbana-Champaign (2008), 92 p.
4. Howard Wainer, (2000). Computerized Adaptive Testing: A Primer. 2 Edition, Routledge.
5. Dang Kh. F., Kamaev V. A., Shabalina O. A., Metod razrabotki algoritmov adaptivnogo testirovaniy, Journal: Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta, 2012, ch. 4, №13.
6. Frank B. Baker, (2001). The Basics of Item Response Theory. 2nd Edition. ERIC Clearinghouse on Assessment and Evaluation.
7. Chaitanya Ekanadham, Yan Karklin (2016). T-SKIRT: Online Estimation of Student Proficiency in an Adaptive Learning System. Available at: https://dsp.rice.edu/sites/dsp.rice.edu/files/tskirt_camera_ready(1)_3.
8. Alekseeva I.V., Gaydey V.O., Dikhovichniy O.O., Konovalova N.R., Fedorova L.B., Statistichniy analiz testovih zavdan iz zastosuvannyam suchasnih matyematichnih modeley, Lviv Polytechnic National University Institutional Repository, 2016. Available at: http://ena.lp.edu.ua
9. Duban R. M., Informatsiyna tekhnologiya pidtrimki testovogo kontrolyu znan zi splayn-modelyami IRT, dissertation abstract: 05.13.06 – informational technology, Lviv, 23 p. (2014).
10. Duban R. M., Splayn-modeli profiliv skladnosti pitan ta znan respondentiv v testovomu kontroli znan, Avtomatizirovannye sistemy upravleniya i pribory avtomatiki: vseukr. mezhvedomstv. nauchn.-tekhn. zb., Kharkiv, 156 (2011).
11. Duban R. M., Ermitiv kubichniy splayn z fiksovanimi krayami – model profiliv IRT, Journal: Komp'yuterno-integrovani tekhnologiї: osvita, nauka, virobnitstvo: naukoviy zhurnal, Lutsk, 2012, № 10, pp. 115-122.
12. Duban R. M., Zastosuvannya integralu vid liniynogo B-splayna v yakosti modeli IRT, Journal: Yelektronika ta sistemi upravlinnya, Kiev, 2012, №1(31), pp. 131-138.
13. Ronald K. Hambleton and Russell W. Jones. (2016). Comparison of Classical Test Theory and Item Response Theory and Their Applications to Test Development. University of Massachusetts at Amherst. Available at: http://www.internationalgme.org/Resources/Pubs/ITEMS_Module_16.

14. Computer adaptive practice of Maths ability using a new item response model for on the fly ability and difficulty estimation S. Klinkenberg, M. Straatemeier, H.L.J. van der Maas Computers & Education 57 (2011) 1813–1824.

15. Wei He Northwest, Qi Diao, Carl Hauser (2016). A Comparison of Four Item-Selection Methods for Severely Constrained CATs. Available at: http://files.eric.ed.gov/fulltext/ED542221.

16. Loken E. Estimation of a Four-Parameter Item Response Theory Model, British Journal of Mathematical and Statistical Psychology, 2010, № 63, pp. 509-525.

17. Assessment systems for good measure (2016). CATSim. Available at: http://www.assess.com/product/catsim/.

18. Vector Psychometric Group (2016). Adaptest®. Available at: https://www.vpgcentral.com/software/computerized_adaptive_testing/.

19. Assessment CenterSM (2016). What is assessment center SM. Available at: https://www.assessmentcenter.net/.

20. WINSTEPS (2016). WINSTEPS & Facets Rasch Software. Available at: http://www.winsteps.com/.

21. CRAN (2016). The Comprehensive R Archive Network. Available at: https://cran.r-project.org/.

22. Seung W. Choi (2016). Firestar: Computerized Adaptive Testing Simulation Program for Polytomous Item Response Theory Models. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.6717 &rep=rep1&type=pdf

23. Package ‘catIrt’ (2016). An R Package for Simulating IRT-Based Computerized Adaptive Tests. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.368.67&rep=rep1&type=pdf.

24. Package ‘catR’ (2016). Generation of IRT Response Patterns under Computerized Adaptive Testing. Available at: https://cran.r-project.org/web/packages/catR/catR.pdf.

25. Package ‘mirtCAT’ (2016). Computerized Adaptive Testing with Multidimensional Item Response Theory. Available at: https://cran.r-project.org/web/packages/mirtCAT/mirtCAT.pdf.

26. Shiny is an RStudio project (2016). Persistent data storage in Shiny ap*ps*. Available at: http://shiny.rstudio.com/articles/persistent-data-storage.html.

27. R. Philip Chalmers, Generating Adaptive and Non-Adaptive Test Interfaces for Multidimensional Item Response Theory Applications: Journal of Statistical Software, July 2016, Volume 71, Issue 5.

28. R. Philip Chalmers mirt: A Multidimensional Item Response Theory Package for the R: Environment Journal of Statistical Software, May 2012, Volume 48, Issue 6. http://www.jstatsoft.org/.