Ministry of Education and Science of Ukraine

Odessa National Academy of Food Technologies

International Competition of Student Scientific Works

BLACK SEA SCIENCE 2020

Information Technology, Automation and Robotics

Proceedings

Odessa, ONAFT 2020

UDC 004.01/08

Editorial board:

Black Sea Science 2020: Proceedings of the International Competition of Student Scientific Works. Information Technology, Automation and Robotics. / Odessa National Academy of Food Technologies; B.Yegorov, M. Mardar, S.Kotlyk (editors-in-chief.) [*et al.*]. – Odessa: ONAFT, 2020. – 365 p.

These materials of International Competition of Student Scientific Works «Black Sea Science 2020» contain the works of the contest participants in the section «Information technologies, automation and robotics» (not winners).

The author of the work is responsible for the accuracy of the information.

Odessa National Academy of Food Technologies, 2020.

Organizing committee:

Prof. Bogdan Iegorov, D.Sc., Rector of Odessa National Academy of Food Technologies, Head of the Committee

Prof. Maryna Mardar, D.Sc., Vice-Rector for Scientific and Pedagogical Work and International Relations of Odessa National Academy of Food Technologies, Deputy Head of the Committee

Prof. Stefan Dragoev, D.Sc., Vice-Rector for Scientific Work and Business Partnerships of University of Food Technologies (Bulgaria)

Prof. Baurzhan Nurakhmetov, D.Sc., First Vice-Rector of Almaty Technological University (Kazakhstan)

Prof. Mircea Bernic, Dr. habil., Vice-Rector for Scientific Work of Technical University of Moldova (Moldova)

Prof. Jacek Wrobel, Dr. habil., Rector of West Pomeranian University of Technology (Poland)

Prof. Michael Zinigrad, D.Sc., Rector of Ariel University (Israel)

Dr. Mei Lehe, Ph.D., Vice-President of Ningbo Institute of Technology, Zhejiang University (China)

Prof. Plamen Kangalov, Ph.D., Vice-Rector for Academic Affairs of "Angel Kanchev" University of Ruse (Bulgaria)

Dr. Alexander Sychev, Ph.D., Assoc. Professor of Sukhoi State Technical University of Gomel (Belarus)

Dr. Hanna Lilishentseva, Ph.D., Assoc. Professor, Head of the Department of Merchandise of Foodstuff of Belarus State Economic University (Belarus)

Prof. Heinz Leuenberger, Ph.D., Professor of the Institute of Ecopreneurship of University of Applied Sciences and Arts (Switzerland)

Prof. Edward Pospiech, Dr. habil., Professor of the Institute of Meat Technology of Poznan University of Life Sciences (Poland)

Prof. Lali Elanidze, Ph.D., Professor of the Faculty of Agrarian Sciences of Iakob Gogebashvili Telavi State University (Georgia)

Dr. V. Kozhevnikova, Ph.D., Senior Lecturer of the Department of Hotel and Catering Business of Odessa National Academy of Food Technologies, Secretary of the Committee

The jury for the section «Information technologies, automation and robotics»

Head of the jury:

Serhiy Kotlyk – Ph.D., Associate Professor, Director of the P.M. Platonov Educational-Scientific Institute of Computer Systems and Technologies "Industry 4.0" of Odessa National Academy of Food Technologies

Members of the jury:

Francisco Augusto – Dr., International Relations Manager of Higher Institute of Information and Communication Technologies (Angola)

Andrey Kuprijanov – Ph.D., Associate Professor of the Department of Software for Computers and Automated Systems of Belarusian National Technical University (Belarus)

Simon Milbert – Vice-President of Xtra Information Management, Inc. (USA)

Ivan Palov – D.Sc., Professor of University of Ruse "Angel Kanchev" (Bulgaria)

Gerard H. Degla – Communications and Training Manager of "MAPCOM solutions informatiques" company group (Benin)

Viktor Khobin – D.Sc., Professor, Head of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies

Valerii Levinskyi – Ph.D., Associate Professor of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies

Viktor Yehorov – Ph.D., Supervisor of the Laboratory of Mechatronics and Robotics of Odessa National Academy of Food Technologies

Valeriy Plotnikov – D.Sc., Professor, Head of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies

Pavlo Lomovtsev – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies

Yurii Kornienko – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies

Sergii Artemenko – D.Sc., Professor, Head of the Department of Computer Engineering of Odessa National Academy of Food Technologies

Serhii Shestopalov – Ph.D., Associate Professor of the Department of Computer Engineering of Odessa National Academy of Food Technologies

Secretary of the jury:

Oksana Sokolova – Senior Lecturer of the Department of Advanced and Applied Mathematics of Odessa National Academy of Food Technologies

TABLE OF CONTENTS

3. Efimenko M.S., Klymiv S.I., Satkenov R.B. Wireless sensor networks // Young Scientist. - 2018. - No. 51. - S. 40-42.

4. Bertrand A. Applications and trends in wireless acoustic sensor networks: A signal processing perspective/ Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

5. Romanova O. V. Speech recognition systems in the modern world [Electronic resourse]. - https://infourok.ru/issledovatelskaya-rabota-na-temu-sistemi-raspoznavaniya-rechi-v-sovremennom-mire-1459208.html

6. Geitgey A. Machine Learning is Fun! The world's easiest introduction to Machine Learning [Electronic resource]. - https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471

7. Kotelnikov V. A. About the bandwidth of ether and wire in telecommunications - All-Union Energy Committee. // Materials for the I All-Union Congress on the Technical Reconstruction of Communication and the Development of Low-Current Industry, 1933. Reprint of the article in UFN, 176: 7 (2006), 762-770.

8. Davis K.P. Forest Fire: Control and Use/ K.P. Davis, G.M. Byram, W.R.Krumm. – N.Y.: McGray-Hill, 1959. – 584 p.

9. Zhuravleva L. V., Strigulin K. A. Studies of the features of the development of neural networks in the modern world [Text] // Technical sciences: problems and prospects: materials of the IV Intern. scientific conf. (St. Petersburg, July 2016). - SPb .: Own publishing house, 2016. - S. 9-11.

10. Le N. V., Panchenko D. Speech recognition based on artificial neural networks [Text]//Technical sciences in Russia and abroad: materials of the Intern. scientific conf. (Moscow, May 2011).- M .: Your printing partner, 2011.-S.8-11.

AN APPLICATION FOR DEMONSTRATING AND COMPARING SORTING AND RETRIEVAL ALGORITHMS

Author: Oleksandr Sokolskyi
Supervisor: Oleksandr Melnykov
Donbass State Engineering Academy (Ukraine)

Abstract. *The use and information and communication of their technology and in education is constantly increasing, including - when teaching professional disciplines in high school. The study of algorithms for sorting and searching data is provided by many educational programs of the specialties of the field of knowledge "Information Technologies". Using the process of presenting these sections of the information and communication means learning how a demonstration is application allows to better understand the essence of each algorithm, cf. ivnyaty them on specific examples.*

The purpose of the work is to develop a program (application) in the visual programming environment, which would allow students studying the algorithms for sorting and searching data , to observe the process and to analyze the advantages and disadvantages of a number of methods to better understand the principles of their work.

The object of study - algorithms for sorting and searching data . The subject of the study is demonstration of a number of algorithms and their comparison.

The first section looks at some algorithms and sorts and retrieves data, and analyzes existing software systems. The second section is performed to develop object-oriented model of a software system means s visual design and UML notation functional model BPWin. The third section provides a user guide and an example of how the software system model is developed.

Key words: *algorithms, sorting, search, composition of the algorithm, computerization, application, uml modeling, functional modeling, functional meling, functional melening.*

I. Introduction

New information technologies occupy the most important place not only in specialized but also in everyday spheres of life. Computers are used in business, management, trade, training and many other areas of human activity. Postiyno growing use and iformatsiyno and communication s technologies and in education , including - in the teaching of professional disciplines of Higher ground school.

The study of algorithms for sorting and searching data is provided by many educational programs of the specialties of the field of knowledge "Information Technologies". Use during the presentation of these sections of the information and communication means learning how a demonstration is application allows to better understand the essence of each algorithm to equalize them with specific examples.

The task was to create a program (application) to demonstrate and compare algorithms for sorting and finding data in order to better understand the principles of their operation.

The object of study - algorithms for sorting and searching data . The subject of the study is demonstration of a number of algorithms and their comparison.

The purpose of the work is to develop a program (application) in the visual programming environment, which would allow students who study sorting and data retrieval algorithms to observe the process and to analyze the advantages and disadvantages of a number of methods. In order to achieve this goal, it is necessary to select algorithms for analysis, to carry out information system design (application) and to carry out its computer implementation.

II Analysis of data collection and search algorithms

Although in the dictionaries the word "sorting" (sorting) is defined as the process of separation of objects by species or variety, programmers have traditionally used it in a very narrow sense, designating them as the rearrangement of objects in which they are arranged in ascending or descending order. This process should probably would not call that sorting and ordering (ordering), but the use of the word would lead to confusion because of congestion word meaning "order." In the following we will use the word "sorting" in the narrow sense: "arrangement in order" [1].

The task of sorting is to find the following permutation of records p (1) p (2) ... p (N) with indexes {1,2, ..., N}, after which the keys would be arranged in descending order:

$$Kp(1) \le Kp(2) \le ... \le Kp(N), \quad (1.1)$$

where Kp (i) is the ith key,

N is the number of entries.

Typically, sorting methods are divided into two classes: internal, when all records are stored in fast RAM, and external, when all records are not in it. Internal sorting

methods provide more flexibility in building data structures and accessing them, while external methods provide the desired result in “spartan” conditions of limited resources.

A fairly good general algorithm spends time sorting N records in proportion to N log N; this requires about log N "passes" according to the data. This is the minimum time if the entries are arranged in random order and sorting is done by pairwise comparison of the keys. If you double the number of records, then the time, all else being equal, increases slightly more than twice. On the other hand, if the keys are known to be random variables with some continuous distribution, then sorting can be performed on average in O (N) steps.

There is a basic list of sorting algorithms:

a) sorting by insertion method. The elements are viewed one at a time, and each new element is inserted into a corresponding place among the previously ordered elements;

b) exchange sorting. If the two elements are not in order, they are interchanged. This process is repeated until the items are ordered;

c) sorting by choice. First, the smallest (or largest) element is selected and in any way separated from the others, then the smallest (largest) is selected from the rest, etc .;

d) sorting by calculation. Each element is compared to all the others; the final position of the element is determined after counting the number of smaller keys;

e) special sorting. It is well suited to the five elements specified in the task, but is not easy to summarize if the elements are larger;

e) a new sorting supermethod. These are substantially advanced known methods.

There are many different sorting algorithms [2], co wives has its advantages and disadvantages, as he is better in some other configuration data and equipment.

In developing our application program we will use the following algorithms: bubble sorting; sorting inserts; sorting by choice; merger sorting; quick sorting; shaker sorting; gnome sorting; sorting Shell; binary sorting; sequential search; binary search.

Merge sort is a sorting algorithm that sorts lists (or other data structures that can only be accessed sequentially, such as streams) [3-4]. Initially, the task is broken down into several smaller tasks. Then these problems are solved by recursive call or directly if their size is small enough. Finally, their solutions are combined, and the solution of the original problem is obtained. The complexity of the algorithm is determined by the formula O (n * log (n)) , where O (n) is a function of the time complexity of the algorithm; n is the number of input data (array elements).

Quick Sort works recursively , repeating the following steps:

a) select the key index and divide the array into two parts. This can be done in different ways, but in this work the array is split in half;

b) transfer all key elements over the right side of the array, and all items less key - to the left. Now the key element is in the correct position - it is larger than any element on the left and less than any element on the right;

c) repeat the first two steps until the array is completely sorted.

Shuffle sorting, or shaker sorting, or Cocktail sort, is a kind of bubble sorting [5]. The boundaries of the working part of the array (ie the part of the array where the movement occurs) are set at the last exchange location at each iteration. The array is viewed alternately from right to left and from left to right. The best case for this sort is the sorted array, the run time is determined by the formula O (n). The worst case for this sorting is the array sorted in reverse order, the running time is determined by the formula.

Gnome sorting is based on the technique used by the ordinary Dutch garden gnome. This is the method by which the garden gnome sorts a line of flower pots. In the other looks at the next and previous garden pots: if they are in the correct order, he steps one pot forward, otherwise he swaps them and steps one pot back. Boundary Conditions: If there is no previous pot, it steps forward; if there is no such pot, he is done.

Sort of a Shell named after American scientist Donald Shell. At its core, this algorithm is an advanced “ Sort Insert ” algorithm . The meaning of the algorithm is to compare not only the elements standing next to each other, but also at some distance. Originally selected for a rock - a gap through which will compare array elements on each iteration [6] . For the first iteration, the step is defined as the result of dividing the number of elements of the array by 2, then the step is divided by 2. That is, it gradually shrinks , and when equal to one , the last comparisons will occur , and the array will be sorted. The best case for this sort - sorted array, Mr. ayhirshyy case for this sort - sorted second backwards array.

The pyramidal sorting method [7] , invented by D. Williams ohm , is an improvement of traditional tree groups. The general idea behind pyramid sorting is that the pyramid is first constructed from the elements of the source array, and then the elements are sorted. We take the last element of the array as the current one. Replace the top (smallest) element of the array and the current one. We sift the current element (it is now the upper one) through the elemental pyramid. Then we take the penultimate element, etc.

Besides sorting, even at dniyeyu and the major processing procedures are structured information search. The search task has attracted a great deal of attention from scientists (programmers) since the dawn of the computer age. Since the 1950s, the problem of finding elements with a certain property in a given set has begun to be solved [8].

Sequential search is the search for the desired entry in a non-sorted list [9] . From, the entire list is viewed before the record is found. It's the simplest m of search algorithms , not very efficient, but it works on a random list.

B inarnyy (or the method of dividing in half) search data can be Apply of Old to the sorted set of elements whose placement is made in the adjacent memory. The essence of this method is as follows: the search begins with the middle element. When comparing a target with an average sorted list item, one of three results is possible: a value equal to, a target value less than a list item, or a target value greater than a list item. In the first, and best of all, the search is complete. In the other two cases, we can drop half of the list. If the target is greater than the middle element, we know that if it is in the list, then it is after that middle element. This is enough to allow us to drop half of the list by comparison.

In this work, the algorithms for sorting and searching data are characterized by the time of work and the number of iterations spent. For a better understanding of their work, you first need to read a summary of each of them in the application program, as well as begin the study from easier to learn algorithms to more complex. Since the amount of memory consumed by the algorithm is not so important for understanding it and described in the help, it will not be further considered in this work.

Next, we will analyze the existing software systems to solve the problem . An application (application) for demonstrating the work of sorting and data retrieval algorithms is created to facilitate understanding of how these algorithms work for the target audience (1st year students).

There are now quite a few websites that demonstrate the sorting and searching of data. Let's look at some of them.

The article "Algorithms and data structures for beginners: sorting" on the site "Tproger" [10] deals with bubble sorting, sorting by inserts, sorting by choice, sorting by merging, quick sorting and swap method for permutation of elements (Fig. 1.1).

Figure 1.1 - Consideration of bubble sorting on the Tproger site

Also on the site are examples of code in C ++ to implement these algorithms. The main disadvantages include: lack of animation and an example of the operation of each of the algorithms for only one arbitrarily filled array.

Another article "Visualization of Sorting Algorithms" on the same site [11] provides the user with an already animated sorting process, but in turn does not have an example code and also allows the user to enter their own array of data to consider for example the best and worst cases (Fig. 1.2).

An excellent example is the article "Insert Sorting" on the habr website [12]. It includes the advantages of the previous two articles, but it has its own major drawback - only the algorithm for sorting inserts and improving them is considered (Fig. 1.3).

The article is about sorting choices on the same site, which has similar drawbacks [13].

Another example of demonstrating sorting algorithms is the article "Sorting in GIFs: 8 Most Popular Algorithms" on the proglib website [14].

Figure 1.2 - Another example of bubble sorting on the Tproger site

Its disadvantages: lack of accuracy of the algorithm description, and work with a single preset array. An example of a quick sorting demo from this site is shown in fig. 1.4.

An interesting example of a demonstration of sorting algorithms is presented in the article "Sorting all the times and peoples" on the habr website [15]. This example is implemented in Excel using macros. An example of work in Fig. 1.5.

The obvious drawback is information overload. But the main drawback of all the examples above, except the last one, is the need for a permanent Internet connection. Therefore, the created application will focus on simplicity and autonomy. Borland Delphi 7 is chosen as the programming language. It will be a great advantage to add multiple interface languages to reach a larger audience.

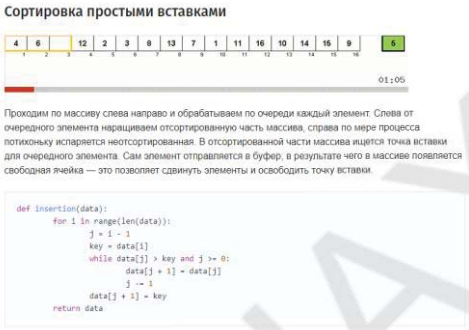

Figure 1.3 - Consideration of sorting with simple inserts on habr site

Figure 1.4 - An example of a quick sort demo on proglib

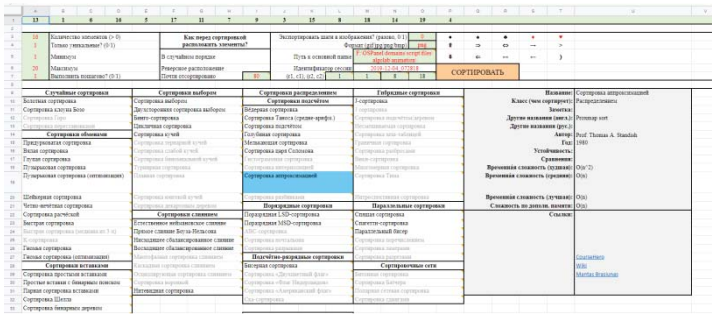

Figure 1.5 - Demonstration of sorting in Excel

III. Design of an application program for demonstration of the algorithms of data sharing and search

In just display will reflect and be only the number of iterations (step algorithm) as real time algorithm too fig s, and the user does not have time to understand the principle along with his work. To solve this problem we must add the delay in following each step of the algorithm, which in turn significantly increase the count time.

Working time is:

$$T = t_i + t_d * N, \quad (2.1)$$

where T is the total running time of the algorithm;

t_i is the time of one iteration;

t_d - delay time;

N is the number of iterations.

When comparing the algorithms, such a problem will not occur, because the size of the array will increase many times, and thus the time of work should be the same. Therefore, the delay can be removed. Then the running time shown will be real.

For a better understanding of user interaction with the application will make the UML- diagram Use Case (Eng. Use case diagram), showing the relationship between actors and precedents, and is part of the model precedent [16].

The following options are available to the actor: working with algorithm demonstration, comparing algorithms and customizing the interface. Working with demonstration algorithms includes : selecting a sort algorithm and / or search algorithm, populating an array, and viewing results. When using the search algorithm, you must specify the desired item.

When comparing algorithms, the user is required to choose the fill option (best and worst case, or random fill) and specify the size of the array that would then analyze the results (by run time and by number of iterations).

Setting up the interface is to choose the language of the software (there are three options: Russian, Ukrainian and English), and if necessary, get help on each of the algorithms for sorting and searching, as well as the whole program as a whole. The diagram of the variants of use is shown in fig. 2.1.

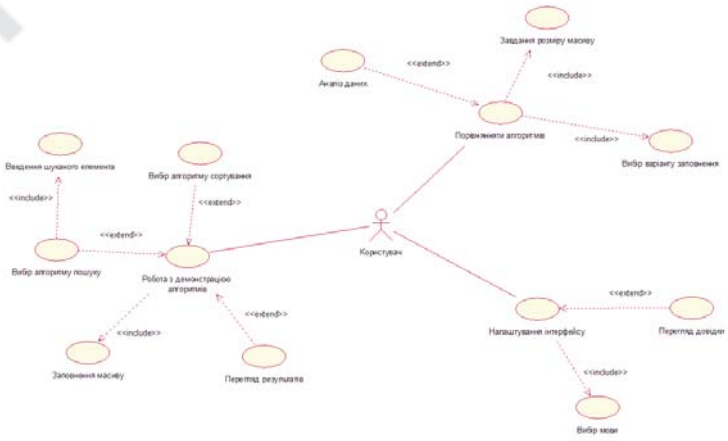

Figure 2.1 - Diagram of use cases

The purpose of the class diagram is a graphical representation of static structures declarative system elements and element s behavior [17]. The constructed diagram of classes is shown in fig. 2.2.

Algorithm Demonstration and Algorithm Comparison classes depend on the Interface class. Communication power of zero or more to one.

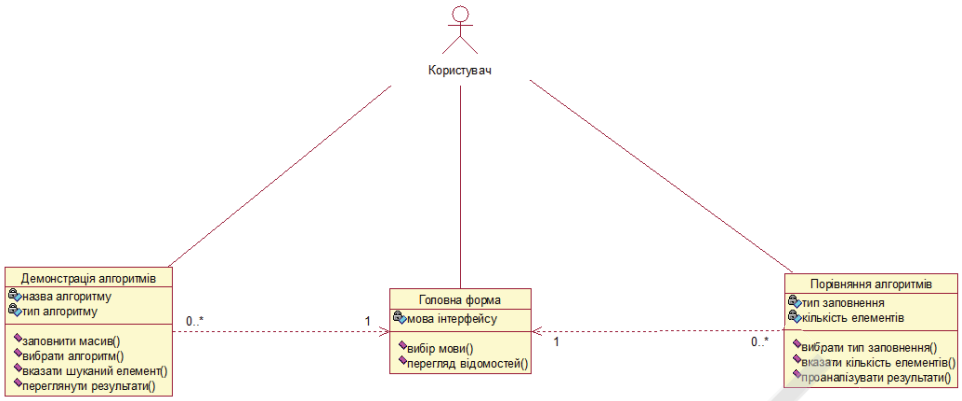

Figure 2.2 - Class diagram

The cooperative diagram [18-19] , which shows the messaging sequence , is presented in Fig. 2.3.

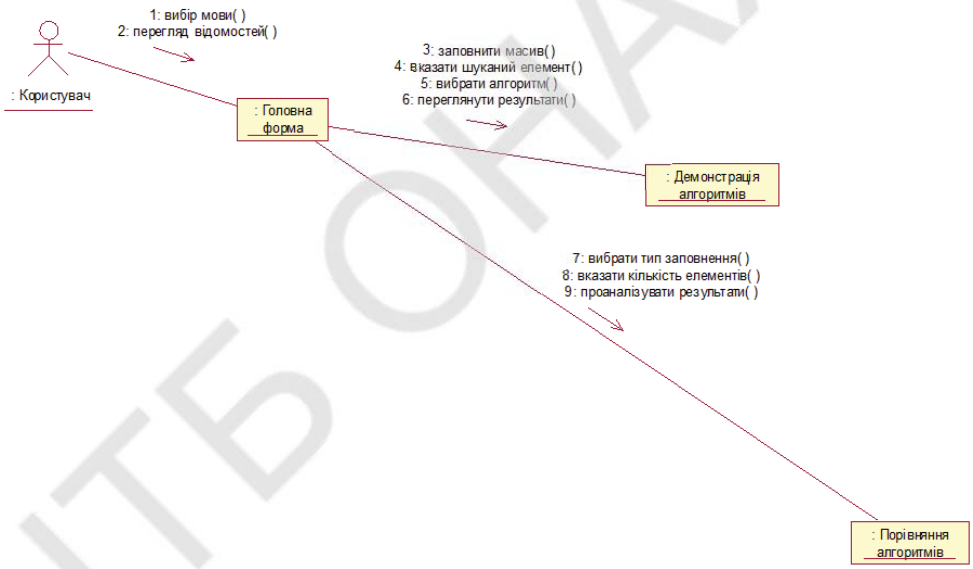

Figure 2.3 - Diagram of cooperation

The component diagram is shown in fig. 2. 4.

In order not to overload the component diagram with unnecessary information, it did not have all the files with Delphi configurations and basic settings. Since the certificate and the application and information on the algorithms loaded with .txt files a total of 36 pieces, for simplification of the diagram were three components Images: «Sort.txt», «Search.txt» and «Program.txt».

Then give functional in a model b system in BPWin notation . The first level of the model in BPWin notation, which shows all input (left) and output (right) information flows, as well as mechanisms (bottom) and governing factors (top) is shown in Fig. 2. 5 .

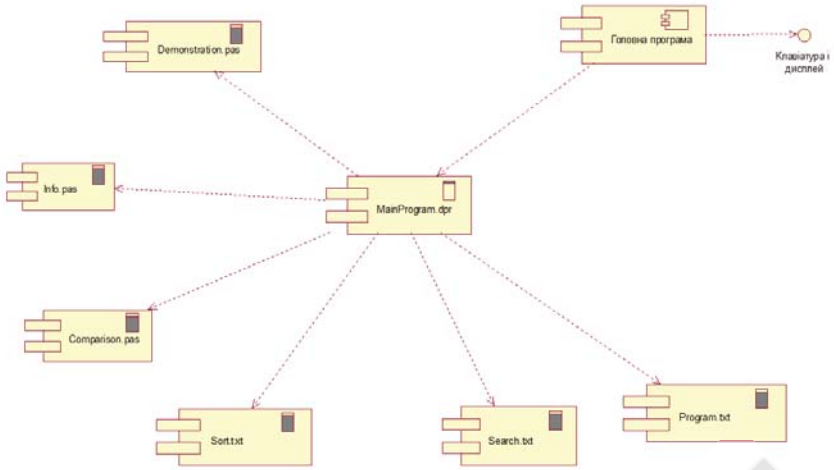

Figure 2. 4 - Component diagram

The second level of the model reflects the three main parts of the software product divided by functional and visual representation on the working forms of the application (Fig. 2.6).

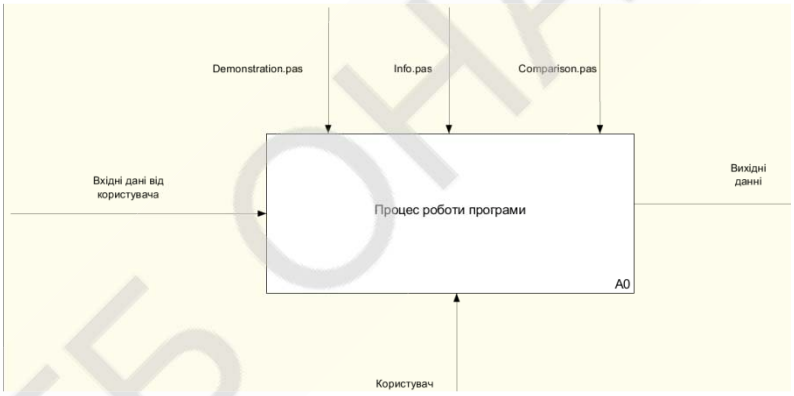

Figure 2. 5 - The first level of the model in BPWin notation

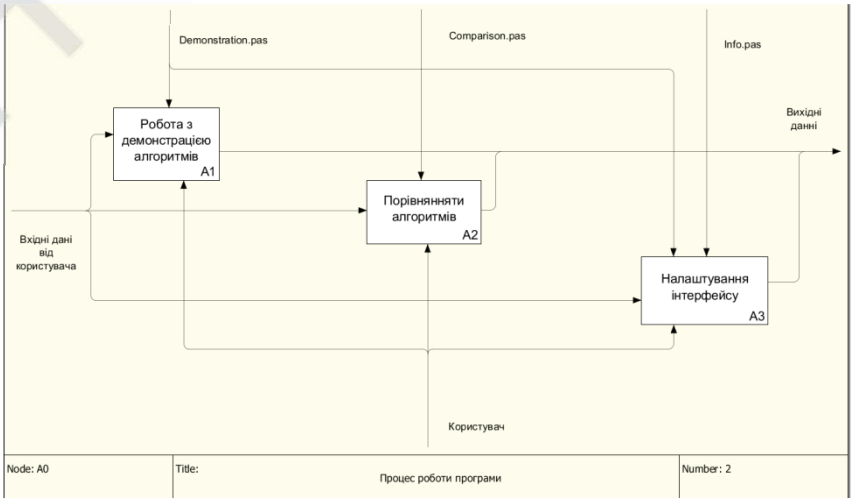

Figure 2. 6 - The second level of the model in BPWin notation

IV. Computer implementation of the software

The scope of the software product is to use it to consolidate knowledge about the principles of sorting algorithms and data retrieval in first year students [20] .

When implementing and using the software, requirements for the functional characteristics, system reliability, hardware parameters, information and software compatibility must be taken into account.

Initially, the main form of the program is developed, which will demonstrate the work of algorithms for sorting and searching data. The form in development is shown in fig. 3.1.

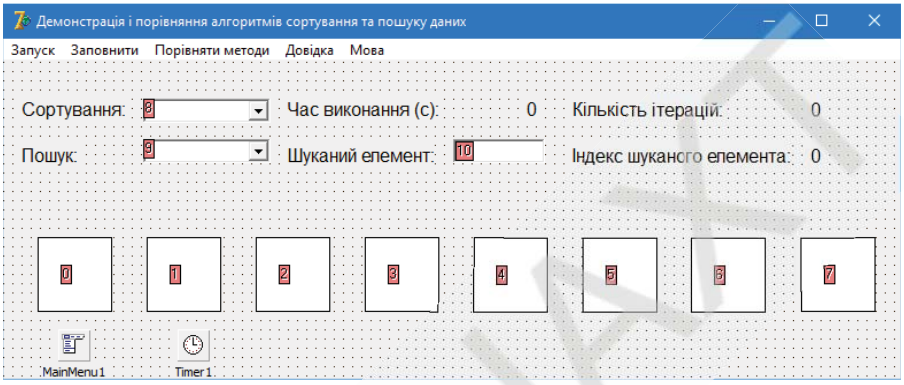

Figure 3.1 - Main form in development

In this form, the user will be able to select sort and search algorithms from the drop-down lists (ComboBox) and set an item to search in an array (Edit field) [21]. Array elements are represented by constituent objects (Edit and Shape). The run time, the number of iterations, and the index of the desired item are output to the appropriate Label component. The transition between forms, interface setup and array filling are represented by the corresponding items in the main menu of the program (MainMenu) [22] . The timer provides the calculation of the running time of the algorithms [23]. In Fig. 3.2 presents the second form of the program - help.

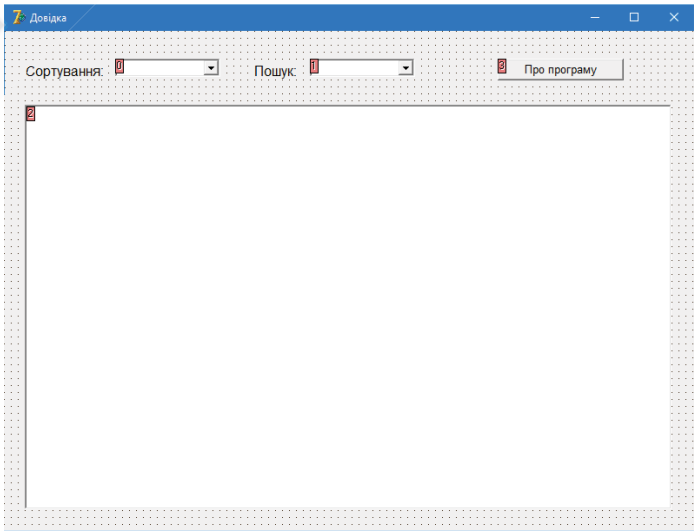

Figure 3.2 - Help window

Similarly, the first form the user can choose the algorithm that it is interested in, and get brief information about it in the Memo. In the same field information about the program is displayed when you press the (Button) button "About". The largest third form for comparing sorting algorithms by run time and the number of iterations spent is presented in Figs. 3.3.

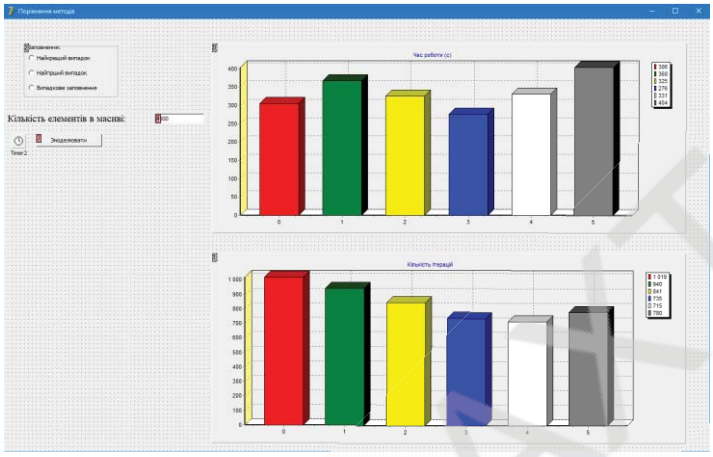

Figure 3.3 - Method Comparison Form

In this form, the user will enter the size of the array (Edit), select the type of fill (RadioGroup) and run the simulation (Button) [24]. Timer provides timing of algorithms and number of iterations. Chart-type elements are used to construct bar charts, to better compare method performance, [25] .

An example of a demonstration of the bubble sorting algorithm program is shown in Fig. 3.4-3.7.

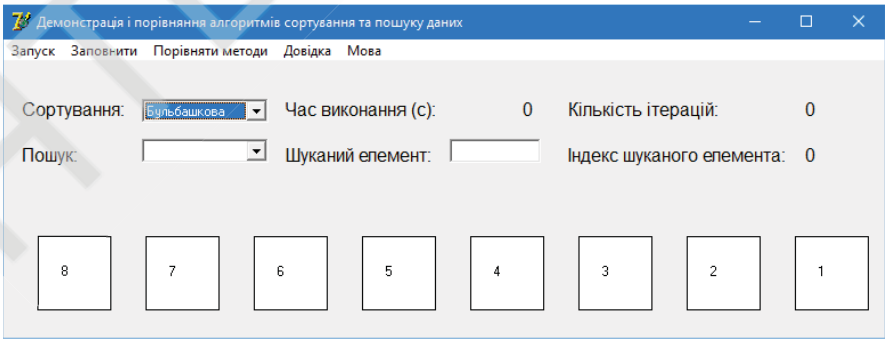

Figure 3.4 - Initial data input

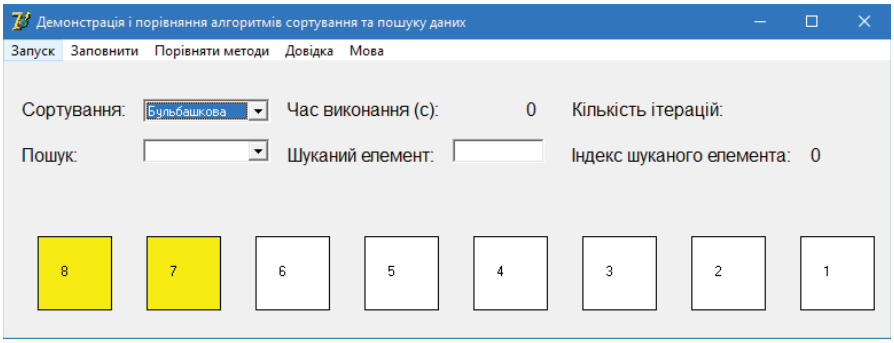

Figure 3.5 - IR procedure verification at x neighboring values

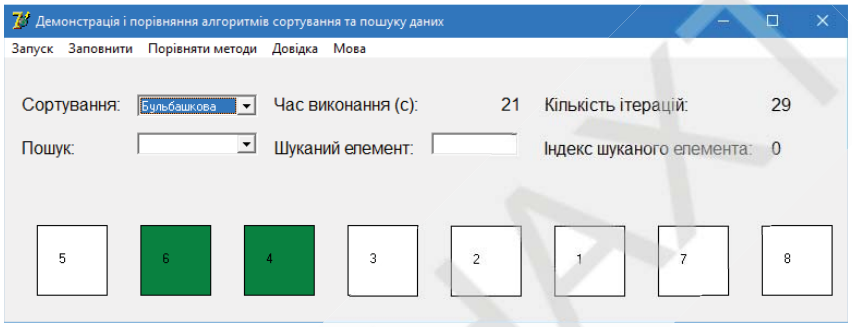

Figure 3.6 - Procedure yard permutation of x Neighbor

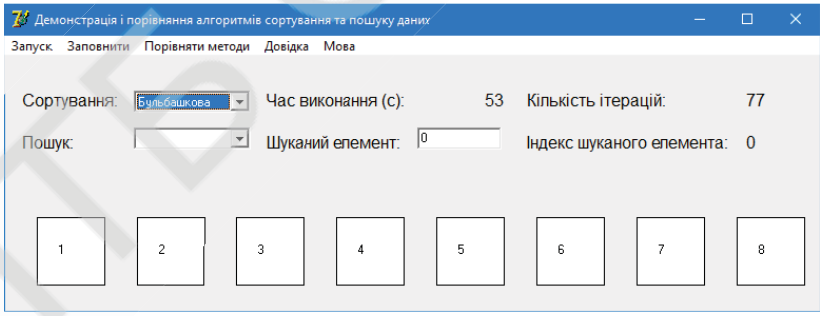

Figure 3.7 - Results of algorithm execution

An example of working with help is the output of information about the sequential search algorithm (Fig. 3.8).

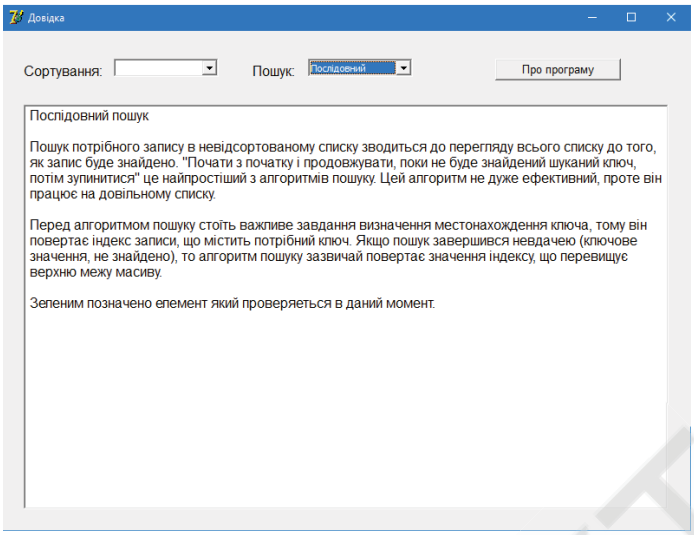

Figure 3.8 - "Serial Search" help information

An example of comparing sorting algorithms in a program by randomly filling an array of 25,000 elements is shown in Fig. 3.9-3.10.

Figure 3. 9 - The process of comparing algorithms

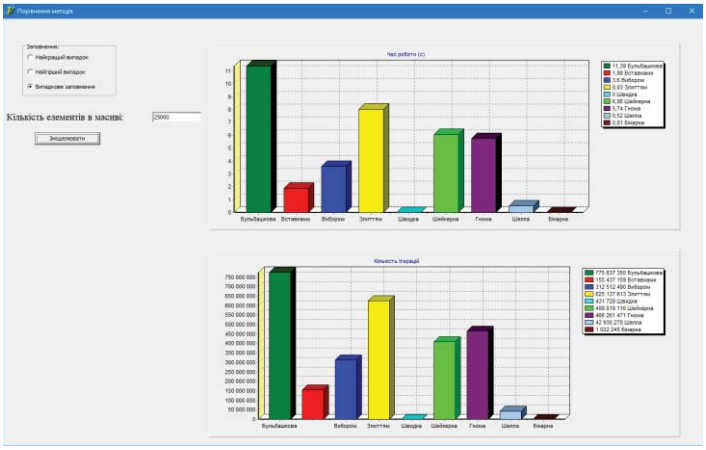

Figure 3.1 0 - Comparison results

An example of changes in the main form when choosing English (Figure 3.1 1):

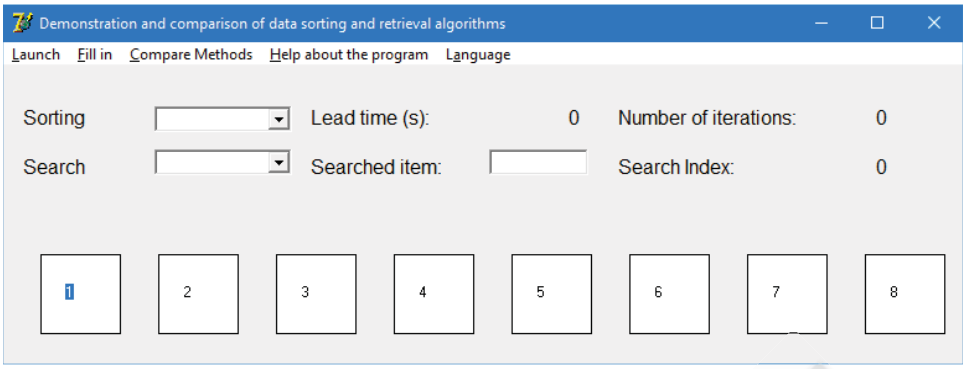

Figure 3.1 1 - The main form of the program in English

Consider the typical sequence of work with the developed application. When you start the program, you open the main form, which houses the main menu of the program , responsible for all the basic functions of the program, array elements and fields for initial values. First of all, you must select the interface language (the fifth item in the main menu). Available and parameters: a) Ukrainian language; b) Russian; c) English.

To demonstrate how the algorithm works, you need to fill in an array and select the sorting and / or search algorithms you are interested in. The array can be filled automatically - the second point Maine st available three options: the best and worst case, random fill. You can manually fill an array with arbitrary integers, select one of the array elements with the mouse and enter the desired value.

After filling in, the desired algorithm is sorted and / or searched from the appropriate list. Available algorithms: bubble sorting; sorting inserts; sorting by choice; merger sorting; quick sorting; shaker sorting; gnome sorting; sorting Shell; binary sorting; sequential search; binary search. If a search algorithm is selected , you must specify the search item in the box to the right of the list of sorting algorithms.

To start the demo, select the first item in the main menu of the application. I n case and elected binary search as the initial data presented are not sorted second by not reducing the array, the program sorts his own "fast" sorting. Time , spent it , will not be taken into account when calculating the algorithm binary search. This is necessary , because binary search can only work with sorted arrays.

Upon completion, the application outputs the running time of the algorithms, the number of iterations, and the index of the search item if the search algorithm was used. If the array does not contain the desired element, the index of the searched element will be null.

To compare sorting algorithms, select the third menu item. A new form will open in which to select from the list one of the possible options for filling and enter the size of the array in the appropriate field. Then the work process is started - pressing the "Model" button. The operation time of the algorithms depends on the computing power of the computer, and when working with large arrays can take quite a long time.

V. Conclusions

The paper analyzes a number of algorithms for sorting and searching data, existing software systems for solving this task. We have designed an application to demonstrate

sorting and data retrieval algorithms using UML and BPWin. The developed information model has received software implementation in Borland Delphi 7 environment.

The created application allows to demonstrate work of a number of algorithms of sorting and search of the data, and also to make their comparison by the criteria "Hours of work" and "Number of iterations". So is the application could be an additional element of information and communication means training in presenting relevant disciplines - for example, "Algorithms and Data Structures" for specialty 124 "System Analysis".

VI. References

1. Melnikov A.Yu. Development of applications for demonstration Sorting algorithms work and claim of the claim data / AJ Melnikov, AS Sokolsky // Automation and computer-integrated technologies in manufacturing and education: status, achievements, prospects materials Ukrainian district and com munities and practical Internet-conference. - Cherkasy, 201 8 . - pp. 204 - 206 .
2. Melnikov A.Yu. Using applications for demonstration work sort algorithms and rovky data and a Search / AJ Melnikov, AS Sokolsky // Modern education - the availability, quality, recognition, collection of Mr. and tific papers international scientific and technical conference, November 14-15, 2018, c. Kp and matorsk / under total. ed. Dr. Techn. Professor, prof. SV Kovalevsky. - Kramatorsk: DDMA , 2018. - P.282-285.
3. Kolyada MG A window into the wonderful world of computer science: a textbook / MG Kolyada. - Donetsk: Stalker, 1997. - 448 p.
4. Khrotko G. High-Level Programming Languages: English Pen. / G. Khrotko - Moscow: MNIIPU, 1982. - 56 p.
5. Tyugashev AA Programming languages: a textbook for universities / AA Tyugashev. - Petersburg: Peter, 2014. - 336 p.
6. Aho AV Data structures and algorithms: trans. with English. / A. V. Aho, D. E. Hopcroft, D. D. Ullman. - Moscow: Williams Publishing House, 2000. - 231 p.
7. Akhtamova S.S. Algorithms for data search: a textbook / SS Akhtamova. - Moscow: Modern high technology technologies, 2007. - 14 p.
8. Korolev LN Informatics. Introduction to Computer Science: A Textbook / LN. Korolev, A.I. Micky. - Moscow: Abris, 2012.- 367 pp.
9. Knut D. The Art of Programming: In 7 Vols. 3. Sorting and Searching: Pen with English. / D. Knut. - Moscow: Williams Publishing House, 2003. - 832 p.
10. Algorithms and data structures for beginners: sorting [Electronic resource]: Tproger . ru . URL : https://tproger.ru/translations/sorting-for-beginners/
11. Visualization of Sorting Algorithms [Electronic resource]: Tproger . ru . URL : https://tproger.ru/digest/sorting-algorithms-visualized/
12. Insert Sorting [Electronic resource]: habr . com . URL : https://habr.com /post/415935/
13. Sort by Choice [Electronic resource]: habr . com . URL : https://habr.com/post/422085/
14. Sorting in GIFs: 8 Most Popular Algorithms [Electronic resource]: proglib . io . URL : https://proglib.io/p/sort-gif/
15. Sorting all the times and peoples [Electronic resource]: habr . com . URL : https://habr.com/post/414447/

16. Melnikov A. Yu. Object-oriented analysis and design of information systems: a textbook / A. Yu. Melnikov. - Ed. 2nd, remaking. and ext. - Kramatorsk: DGMA, 2013. - 172 p.
17. Leonenko AV The UML Tutorial / AV Leonenko. - 2nd ed., Remaking. and ext. - St. Petersburg: BHC-Petersburg, 2004. - 432 p.
18. Butch Grady. UML language. User Manual / Grady Butch, James Rambo, Aivar Jacobson; trans. with English. A.A. Slinkin. - 2nd ed., P. - M .: DMK Press; St. Petersburg: Peter, 2004. - 432 p.
19. Schmuller Joseph. Master UML yourself in 24 hours / Joseph Schmuller. - 2nd edition; trans. with English. - M .: Williams Publishing House, 2002. - 352 p.
20. Sokolsky OS Use of special application of own development for demonstration of comparison of algorithms of data sorting / O.C. Sokolsky, O.Yu. Melnikov // Proceedings of the II All-Ukrainian scientific-practical Internet conference of students, graduate students and young scientists on the topic "Modern computer systems and networks in management": a collection of scientific works / Edited by G.O. Raiko. - Kherson: VF Vyshemirsky FOP, 2019. - P.378-380. - ISBN 978-617-7783-32-8 (online edition)
21. Melnikov A. Yu. Work in Borland-Delphi environment: A training manual for students of the specialty "Economic cybernetics" / A. Yu. Melnikov. - Kramatorsk: DGMA, 2004. - 80 p.
22. Goffman V.E. Delphi 5 / V.E. Hoffman, A. D. Khomonenko. - St. Petersburg: BHC - St. Petersburg, 1999. - 800 p.
23. Darakhvelidze P. Programming in Delphi 7 / P. Darakhvelidze, E. Markov - St. Petersburg: BHC - St. Petersburg, 2000. - 784 p.
24. Halyseev G. V. The components in Delphi 7 / H. W. Halyseev - M .: Williams, 2004. - 624 p.
25. Faronov VV The Delphi Programming System / VV Faronov. - St. Petersburg: BHC-Petersburg, 2003. - 912 p.

INFORMATION SYSTEM FOR WORKING WITH EDUCATION PROGRAMS AND HIGHER EDUCATION STANDARDS

Author: Kateryna Didevych
Supervisor: Oleksandr Melnykov
Donbass State Engineering Academy (Ukraine)

Abstract. *Higher education standard is a set of norms that establish the main purpose and objectives of education, requirements for the content of education, the level of training of specialists, determine the way to diagnose the quality of higher education. Each institution of higher education, on the basis of an approved standard, develops, for each specialty, an educational (vocational, educational or scientific) program that can be approved or modified annually. Creating educational programs based on existing standards, comparing and improving them is a creative process, but the need to constantly update materials and conduct quality assessments requires the involvement of information systems and technologies.*