

- [7] Pavlova S.I. (2016) Project-oriented organizations as the development of enterprise management methods. *Visnyk ZhDTU. Seriya: Ekonomichni nauky.* **4 (78)**. P. 170-176. (in Ukrainian).
- [8] Xlobystova O.A, Hladka M.V. (2013) IT project management: Laboratory workshop for students majoring in 6.050101 "Computer Science" full-time and part-time education. Kyiv: NUXT, 108 p. (in Ukrainian).
- [9] Open Plan and others: the Welcom family of project management software packages. (2008) URL: <https://blog.iteam.ru/open-plan-i-drugie-semejstvo-programmyh-paketov-welcom/> (Accessed date: 06.01.2020).

*Kachan Hanna*

## FEATURES OF IT-PROJECT MANAGEMENT COURSE IN HIGHER EDUCATION INSTITUTIONS

**Abstract.** In today's conditions of development of the information economy, the priority area is the development in the field of information technologies. Software and software development are a complex project that requires specialized project managers to manage. As a result, there is a severe lack of formalization, preparation, and project management skills in the professional environment. Therefore, there is an urgent need to master the global experience of developing, analyzing, implementing, and managing projects, especially in the field of information systems and software development.

The article deals with the peculiarities of the discipline "IT Project Management" in higher education institutions of I-II accreditation level. The main purpose of teaching the discipline "Project Management" is defined as the formation of students' ideas about the methodology of preparation and implementation, ways and means of project construction, attracting resources for the implementation of these projects and mechanisms for managing them. The objective of the course is to give students a system of theoretical knowledge and to develop practical skills in the field of planning, evaluation, monitoring and support of projects in the IT field.

The content of the course "IT Project Management", which is designed to prepare students in the specialty "Computer Science" of all forms of training, is disclosed. Students acquire a complex of knowledge and skills that will allow them to understand the process of project management, to practically use the means of planning and implementation of IT projects, to develop the right structure of the project implementation plan, to acquire practical skills of using information systems in which the development and implementation of project implementation stages, assigning responsible for stages of implementation, to write the terms of reference using the regulatory documents and relevant GOST.

Some methodological aspects of project management training are considered as a necessary link of professional training of specialists in the field of information technologies. A brief overview of the most common project management systems: MS Project, OpenPlan Professional and Project Expert. The course training should facilitate students' acquisition of basic knowledge of the principles of project management theory in software development and the acquisition of laboratory skills in planning, controlling, and optimizing software development processes.

**Keywords:** management, project, IT project, information technology, information systems, software, training program.

DOI 10.31392/NPU-nc.series 2.2020.22(29).11

УДК 004.42:519.6

**Олександр Юрійович Мельников<sup>1</sup>, Олександр Сергійович Сокольський<sup>2</sup>**

Донбаська державна машинобудівна академія, м. Краматорськ, Україна

<sup>1</sup>кандидат технічних наук, доцент, доцент кафедри інтелектуальних систем прийняття рішень

ORCID ID: 0000-0003-2701-8051

*alexandr@melnikov.in.ua*

<sup>2</sup>студент спеціальності «Системний аналіз»

ORCID ID: 0000-0002-3933-5057

*SokolMag2380@gmail.com*

## СПЕЦІАЛЬНЕ ЗАСТОСУВАННЯ ВЛАСНОЇ РОЗРОБКИ ДЛЯ ДЕМОНСТРАЦІЇ І ПОРІВНЯННЯ АЛГОРИТМІВ СОРТУВАННЯ ТА ПОШУКУ ДАНИХ

**Анотація.** У статті наведено опис спеціального застосування власної розробки, яке дозволяє студентам, що вивчають алгоритми сортування та пошуку даних, спостерігати за процесом і

проводити аналіз переваг і недоліків низки методів для кращого розуміння принципів їх функціонування. Розглянуто деякі алгоритми сортування та пошуку даних, проаналізовано існуючі програмні системи (інтернет-сайти) для розв'язування поставленої задачі, їх особливості, переваги та недоліки. Виконано розробку об'єктно-орієнтованої моделі програмної системи з використанням засобів візуального моделювання UML (подані діаграми варіантів використання і діаграма класів) та функціональної моделі в нотації BPWin (перший і другий рівні). Перераховано доступні алгоритми: бульбашкове сортування, сортування вставками, сортування вибором, сортування злиттям, швидке сортування, шейкерне сортування, сортування гнома, сортування Шелла, бінарне сортування, послідовний пошук, бінарний пошук. Оскільки реальний час роботи алгоритму на сучасному комп'ютері є занадто малим, і користувач не встигне зрозуміти принципів його роботи, було прийнято рішення додати затримку після кожного кроку, що істотно збільшило час під час демонстрації роботи алгоритму. Наведено приклади використання розробленого застосування: введення даних і демонстрація роботи алгоритму сортування обмінами, порівняння алгоритмів сортування для випадкового заповнення масиву з 25000 елементів за критеріями «Час роботи» і «Кількість ітерацій». Відзначено можливість зміни мови інтерфейсу. Описано використання довідкової системи. Розглянуто типову послідовність роботи з розробленим застосуванням. Зроблено висновок, що створене застосування може стати додатковим елементом інформаційно-комунікативних засобів навчання при викладенні відповідних дисциплін – наприклад, «Алгоритми і структури даних» для спеціальностей галузі 12 «Інформаційні технології».

**Ключові слова:** алгоритми сортування та пошуку, демонстрація роботи, складність алгоритму, UML-моделювання, функціональна модель, програмне забезпечення, застосування.

Вивчення алгоритмів сортування та пошуку даних передбачено в багатьох освітніх програмах для спеціальностей галузі знань «Інформаційні технології». Використання в процесі навчання цих розділів такого інформаційно-комунікативного засобу навчання, як демонстраційне застосування, дозволяє краще зрозуміти суть кожного алгоритму, порівняти їх на конкретних прикладах.

Зараз існує досить багато сайтів, на яких демонструють застосування алгоритмів сортування та пошуку даних. Розглянемо деякі з них.

На сайті «Tproger» [1] розглянуті бульбашкове сортування, сортування вставлянням, за вибором, злиттям, швидке і swap. Так само на сайті наведені приклади коду на мові C++ для реалізації даних алгоритмів. До головних мінусів відносяться: відсутність анімації і приклад застосування кожного з алгоритмів тільки для одного довільно заповненого масиву. В іншій статті «Візуалізації алгоритмів сортування» на цьому самому сайті [2] надаються користувачеві вже анімований процес сортування, але в свою чергу немає прикладів коду.

В розділі «Сортування вставками» на сайті habr [3] містяться переваги двох попередніх, але є свій власний великий недолік – розглядається тільки алгоритм сортування вставлянням та його вдосконалення. В розділі, присвяченому сортуванню за вибором на цьому самому сайті, наявні аналогічні недоліки [4].

Іншим прикладом демонстрації алгоритмів сортування є «Сортування в гіфках: 8 найбільш поширених алгоритмів» на сайті proglib [5]. Її недоліки: відсутність точності опису алгоритму, робота з єдиним наперед заданим масивом.

Приклад «Сортування усіх часів та народів» на сайті habr [6] реалізований в Excel із використанням макросів. Очевидний недолік – перевантаженість різними відомостями.

Але головний недолік більшості розглянутих вище прикладів – це необхідність наявності постійного під'єднання до мережі Інтернет. Поширеного автономного застосування знайти не вдалося.

Було поставлено завдання створення програми (застосування) для демонстрації та порівняння алгоритмів сортування та пошуку даних з метою кращого розуміння принципів їх функціонування цільовою аудиторією (студентами 1-го курсу). Створювана прикладна програма бути націлена на простоту і автономність.

Раніше авторами було створено застосування для демонстрації алгоритмів сортування та пошуку даних з метою кращого розуміння принципів їх функціонування [7-8]. Користувачеві був доступний інтуїтивно зрозумілий мінімум необхідних елементів управління для виконання всіх функцій програми: списки, що випадають, для вибору алгоритму сортування або пошуку даних; поле для введення шуканого елемента (для пошуку); головне меню. Вхідний масив може бути заповнений за трьома способами: найкращий випадок, найгірший випадок, а також випадкове заповнення значеннями від 0 до 99. Крім автоматичного заповнення за допомогою меню програми, користувач може заповнити масив самостійно будь-якими значеннями. Через реалізоване в середовищі візуального програмування застосування можна побачити процес сортування за дев'ятьма

алгоритмами і процес пошуку за двома алгоритмами [9]. Але це застосування не мало можливості провести ще й порівняння алгоритмів.

Для демонстрації роботи має відображатися тільки кількість ітерацій (кроків алгоритму), тому що реальний час виконання алгоритму занадто малий, і користувач не встигне зрозуміти принципів його роботи. Для розв'язування даної проблеми треба додати затримку після кожного кроку алгоритму, що суттєво збільшить час виконання, який складає:

$$T = t_i + t_z * N, \tag{1}$$

де  $T$  – загальний час виконання алгоритму;

$t_i$  – час виконання однієї ітерації;

$t_z$  – час затримки;

$N$  – кількість ітерацій.

Під час порівняння алгоритмів такої проблеми не виникне, тому що розмір масиву багаторазово зростає, і час роботи отже так само. Тому затримку можна буде прибрати, і показаний час виконання буде відповідати реальному.

Для кращого розуміння дії користувача з застосуванням складемо UML-діаграму варіантів використання (англ. Use case diagram), що відображає відносини між авторами і прецедентами і є складовою частиною моделі прецедентів [10].

Авторові доступні такі варіанти використання: робота з демонстрацією алгоритмів, порівняння алгоритмів і налаштування інтерфейсу. Робота з демонстрацією алгоритмів містить: вибір алгоритму сортування та/або алгоритму пошуку, заповнення масиву і перегляд результатів. Під час використання алгоритму пошуку слід вказати шуканий елемент. Під час порівняння алгоритмів користувач зобов'язаний вибрати варіант заповнення і вказати розмір масиву, щоб потім провести аналіз результатів (за часом виконання і за кількістю ітерацій).

Налаштування інтерфейсу полягає у виборі мови для роботи з програмним продуктом (російська, українська, англійська), і в разі потреби – отримання довідки щодо кожного з алгоритмів сортування та пошуку, а так само всієї програми в цілому. Побудована діаграма варіантів використання наведена на рис. 1.

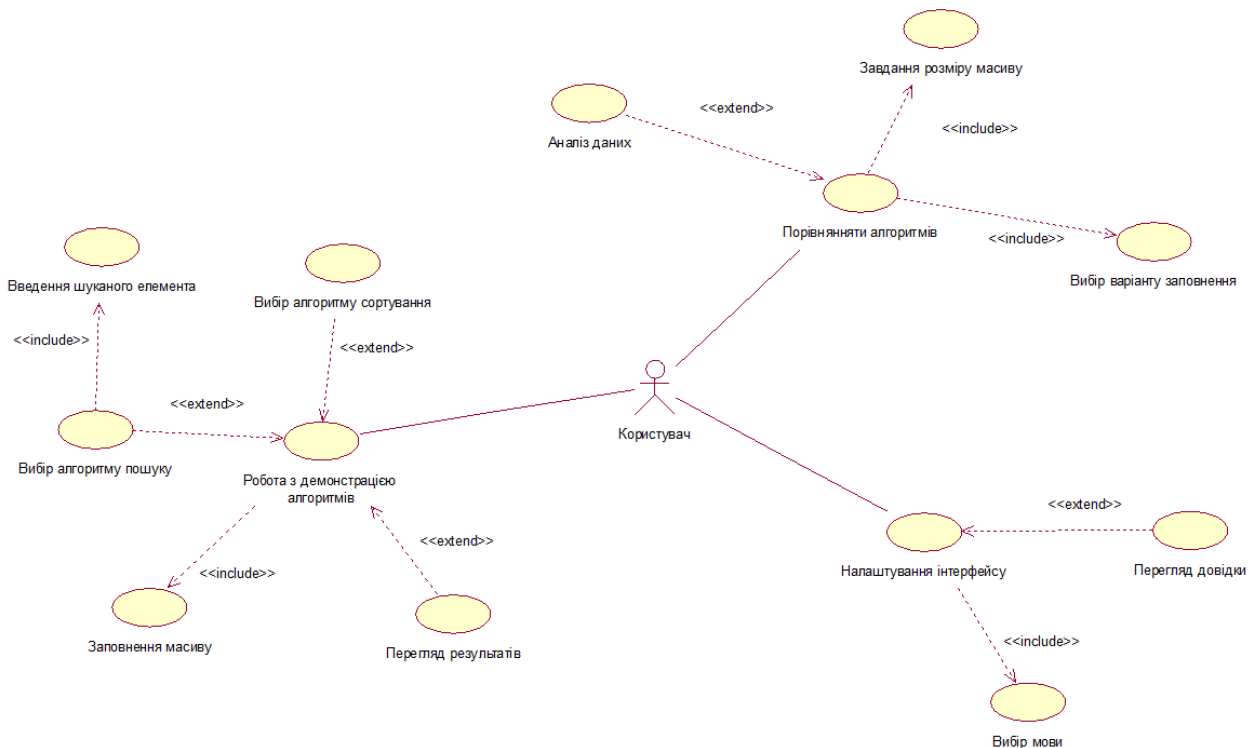


Рис. 1. Діаграма варіантів використання

Метою створення діаграми класів є графічне подання статичної структури декларативних елементів системи та елементів поведінки. Побудована діаграма класів зображена на рис. 2. Класи «Демонстрація алгоритмів» і «Порівняння алгоритмів» залежні від класу «Інтерфейс».

Далі наведемо функціональну модель системи в нотаціях BPWin. Перший рівень моделі в нотації BPWin, на якому показані усі вхідні (ліворуч) та вихідні (праворуч) потоки даних, а також механізми (знизу) та фактори управління (зверху) зображений на рис. 3.

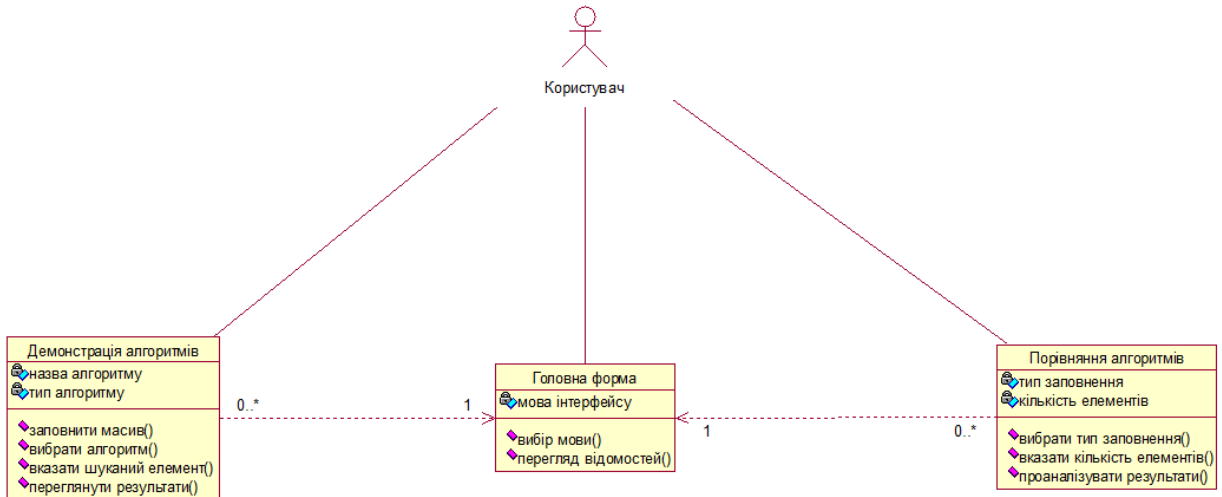


Рис. 2. Діаграма класів

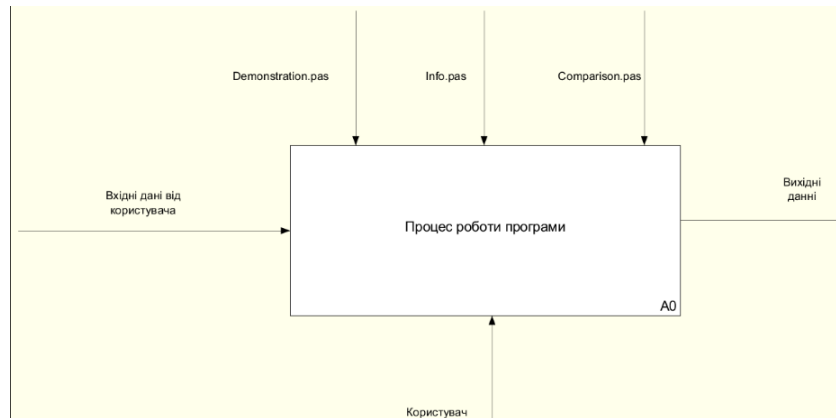


Рис. 3. Перший рівень моделі в нотації BPWin

На другому рівні моделі відображаються три основні частини програмного продукту, розділені за функціоналом та візуальним поданням на робочих формах застосування (рис. 4). Приклад демонстрації створеного алгоритму бульбашкового сортування зображено на рис. 5 – 8.

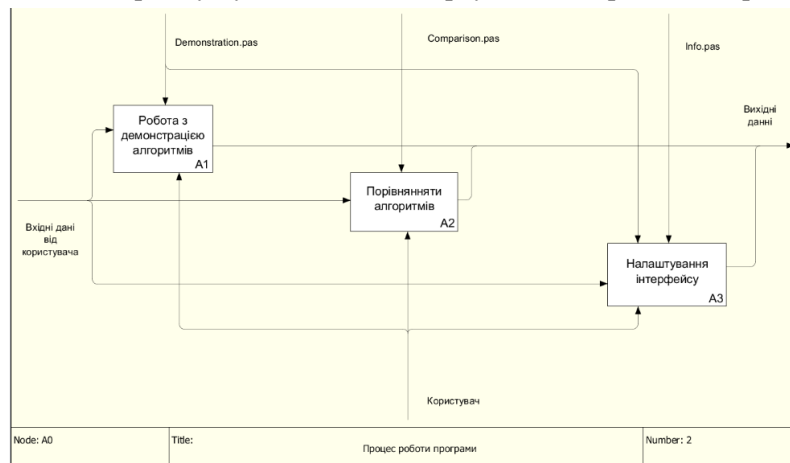


Рис. 4. Другий рівень моделі в нотації BPWin

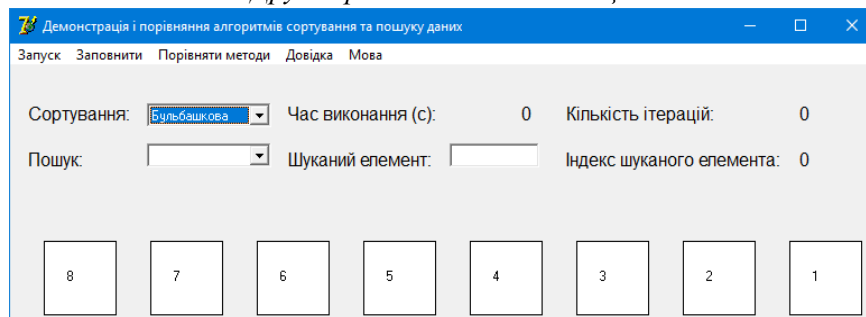


Рис. 5. Введення початкових даних

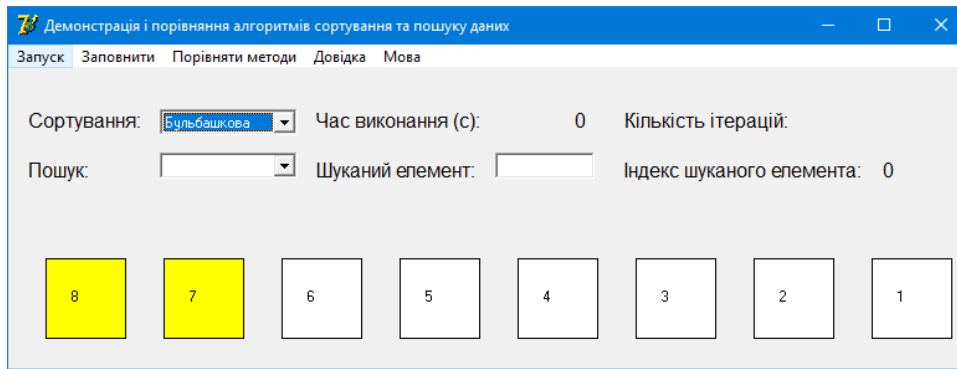


Рис. 6. Процедура перевірки двох сусідніх значень

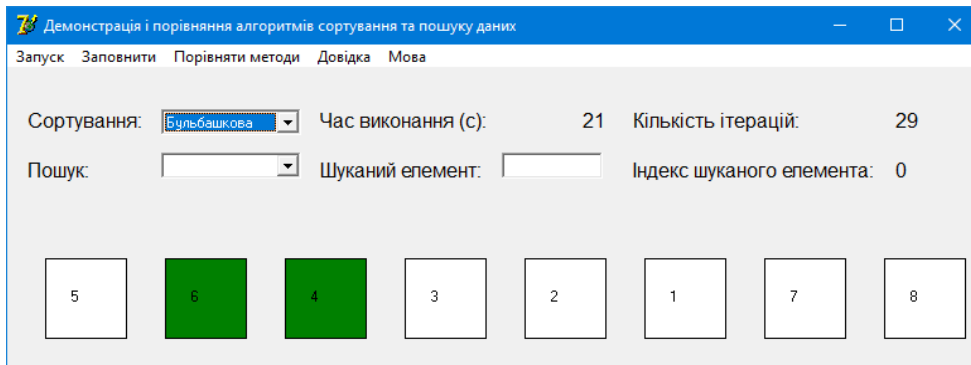


Рис. 7. Процедура перестановки двох сусідніх елементів

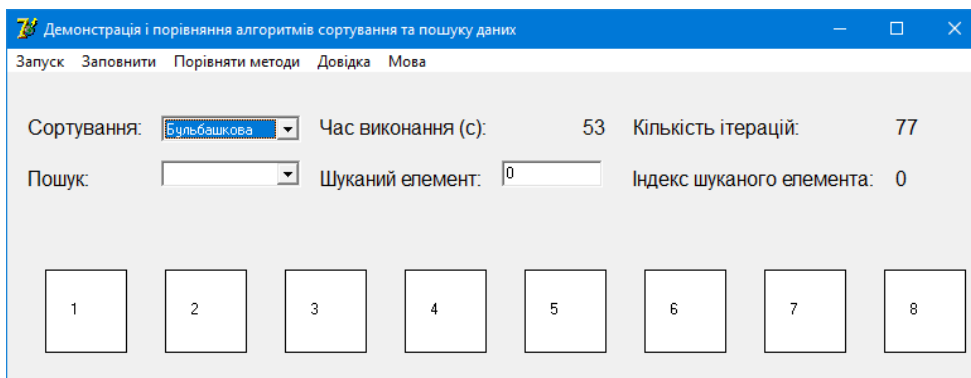


Рис. 8. Результати виконання алгоритму

Приклад роботи з довідкою – відомості про алгоритм послідовного пошуку (рис. 9).

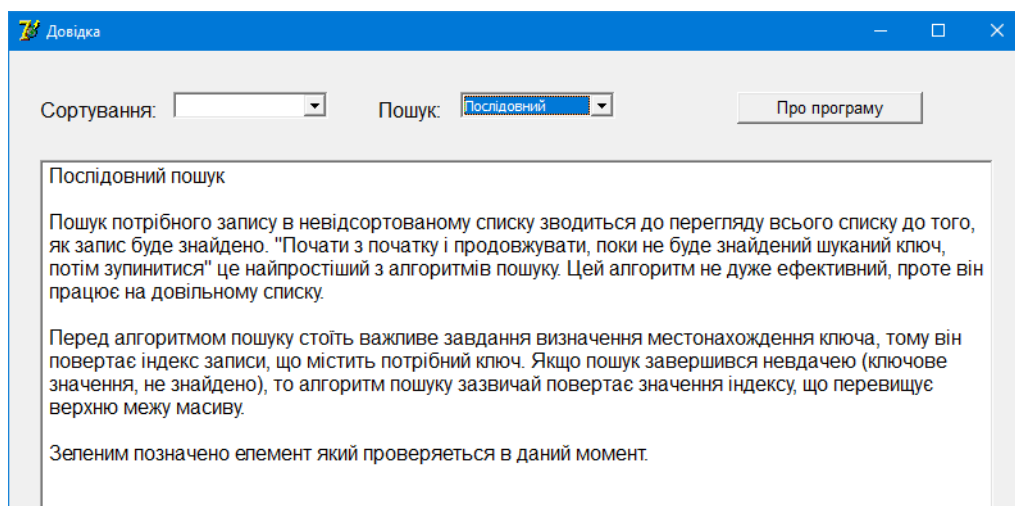


Рис. 9. Довідка про «Послідовний пошук»

Приклад порівняння алгоритмів сортування для випадкового заповнення масиву з 25000 елементів наведено на рис. 10 – 11.



Рис. 10. Процес порівняння алгоритмів

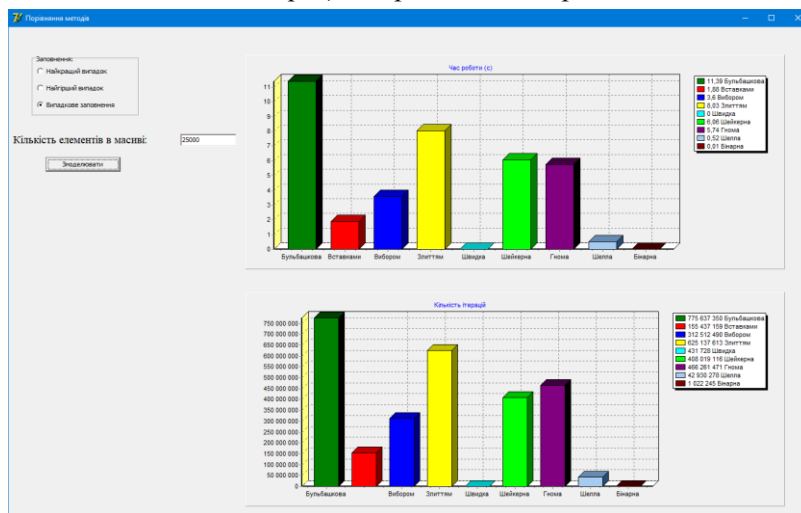


Рис. 11. Результати порівняння

Приклад змін головної форми, якщо користувач обере англійську мову (рис. 12):

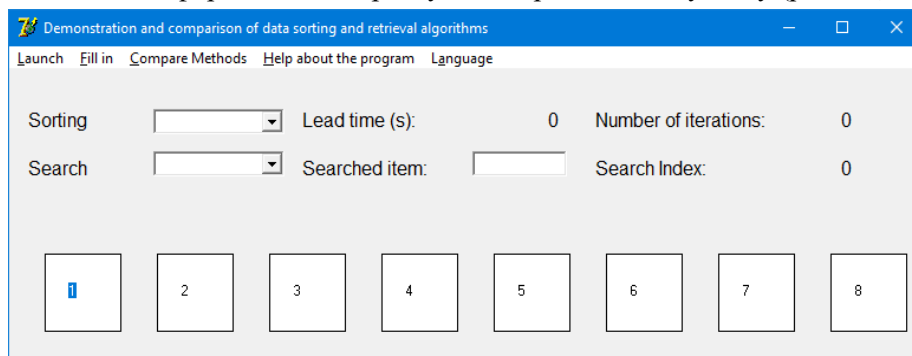


Рис. 12. Головна форма програми англійською мовою

Розглянемо типову послідовність роботи із розробленим застосуванням. Після запуску програми відкривається головна форма, на якій розташовано головне меню програми, де показано всі основні функції програми, елементи масиву і поля для початкових значень.

Для демонстрації роботи алгоритму необхідно заповнити масив і вибрати алгоритми сортування та/або пошуку. Масив можна заповнити автоматично – другий пункт меню, доступні три варіанти: найкращий і найгірший випадок, випадкове заповнення. Можна вручну заповнити масив довільними цілими числами, для цього слід вказати на один з елементів масиву курсором мишки і ввести потрібне значення.

Після заповнення вибирається потрібний алгоритм сортування та / або пошуку з відповідного списку. Доступні алгоритми: бульбашкове сортування; сортування вставлянням; сортування вибором; сортування злиттям; швидке сортування; шейкерне сортування; сортування гнома; сортування Шелла; бінарне сортування; послідовний пошук; бінарний пошук.

Якщо був обраний алгоритм пошуку, слід вказати шуканий елемент в полі праворуч від списку

алгоритмів сортування. Для запуску демонстрації слід вибрати перший пункт головного меню програми.

Якщо обрано бінарний пошук і в якості вихідних даних подано не відсортований масив, за програмою автоматично він відсортується «швидким» сортуванням. Час, витрачений на це, не буде враховуватись під час підрахунку часу виконання алгоритму бінарного пошуку. Це необхідно, тому що бінарний пошук може використовуватись тільки з відсортованими впорядкованими масивами.

Після закінчення роботи за програмою видається час виконання алгоритмів, кількість ітерацій і індекс шуканого елемента, якщо був задіяний алгоритм пошуку. У разі коли в масиві немає шуканого елемента, в якості індексу шуканого елемента буде видано нуль.

Для порівняння алгоритмів сортування слід вибрати третій пункт меню. Відкриється нова форма, в якій слід вибрати зі списку один з можливих варіантів заповнення масиву даних і ввести розмір масиву у відповідне поле. Потім проводиться запуск процесу виконання алгоритму – натисканням кнопки «Змоделювати».

Час роботи алгоритмів залежить від обчислювальної потужності ЕОМ, і робота з великими масивами може зайняти досить багато часу. В процесі опрацювання даних результати виконання алгоритму будуть виводитись в дві стовпчасті діаграми: час роботи – верхня, і кількість ітерацій – нижня.

Для перегляду сутності кожного з алгоритмів сортування та пошуку даних, які використовуються в даній програмі, слід звернутися до довідки – четвертий пункт головного меню програми. Відкриється нове вікно програми, де можна вибрати потрібний алгоритм зі списку, як у головній формі програми. Опис буде показано у вікні в нижній частині форми.

Коротка довідка про програму розташована на цій самій формі. Для її перегляду слід натиснути кнопку «Про програму».

**Висновки.** Через розроблене застосування візуалізується функціонування низки алгоритмів сортування й пошуку даних, а також проводиться їх порівняння за критеріями «Час роботи» й «Число ітерацій». Таке застосування може стати додатковим елементом інформаційно-комунікативних засобів навчання під час навчання відповідних дисциплін – наприклад, «Алгоритми і структури даних» для спеціальності 124 «Системний аналіз».

#### Список використаних джерел

- [1] Алгоритмы и структуры данных для начинающих: сортировка. URL: <https://tproger.ru/translations/sorting-for-beginners/> (дата звернення: 01.11.2019).
- [2] Визуализации алгоритмов сортировки. URL: <https://tproger.ru/digest/sorting-algorithms-visualized/> (дата звернення: 01.11.2019).
- [3] Сортировки вставками. URL: <https://habr.com/ru/post/415935/> (дата звернення: 01.11.2019).
- [4] Сортировки выбором. URL: <https://habr.com/ru/post/422085/> (дата звернення: 01.11.2019).
- [5] Сортировки в гифках: 8 самых популярных алгоритмов. URL: <https://proglib.io/p/sort-gif/> (дата звернення: 01.11.2019).
- [6] Сортировки всех времён и народов. URL: <https://habr.com/ru/post/414447/> (дата звернення: 01.11.2019).
- [7] Мельников А. Ю., Сокольский А. С. Разработка приложения для демонстрации работы алгоритмов сортировки и поиска данных. *Автоматизация та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку*: матеріали Всеукраїнської науково-практичної Internet-конференції. Черкаси, 2018. С. 204-206.
- [8] Мельников А. Ю., Сокольский А. С. Использование приложения для демонстрации работы алгоритмов сортировки и поиска данных. *Сучасна освіта – доступність, якість, визнання*: збірник наукових праць міжнародної науково-методичної конференції (м. Краматорськ, 14–15 листопада 2018 року) / під заг. ред. д-ра техн. наук., проф. С. В. Ковалевського. Краматорськ: ДДМА, 2018. С. 282-285.
- [9] Седжвик Р. Фундаментальные алгоритмы на С. Анализ. Структуры данных. Сортировка. Поиск. Алгоритмы на графах. СПб: ООО «ДиаСофтЮП», 2003. 1136 с.
- [10] Мельников А. Ю. Объектно-ориентированный анализ и проектирование информационных систем: учебное пособие. Изд. 2-е, перераб. и доп. Краматорск: ДГМА, 2013. 172 с.

#### References

- [1] Algorithms and Data Structures for Beginners: Sorting. Available at: <https://tproger.ru/translations/sorting-for-beginners/>. (in Russian)
- [2] Sorting Algorithm Visualizations. Available at: <https://tproger.ru/digest/sorting-algorithms-visualized/>. (in Russian)

- [3] Sort by inserts. Available at: <https://habr.com/ru/post/415935/>. (in Russian)
- [4] Selection sorting. Available at: <https://habr.com/ru/post/422085/>. (in Russian)
- [5] GIF sorts: 8 most popular algorithms. Available at: <https://proglib.io/p/sort-gif/>. (in Russian)
- [6] Sorts of all times and peoples. Available at: <https://habr.com/ru/post/414447/>. (in Russian)
- [7] Mel'nikov A. Ju., Sokol'skij A. S. (2018). Development of an application to demonstrate the operation of algorithms for sorting and searching data. *Automation and computer-integrated technologies in production and education: state, achievements, development prospects: materials of the All-Ukrainian scientific-practical Internet-conference*. Cherkasy. P. 204-206. (in Russian)
- [8] Mel'nikov A. Ju., Sokol'skij A. S. (2018) Using the application to demonstrate the work of algorithms for sorting and searching data. *Modern education – accessibility, quality, recognition: Proceedings of the International Scientific and Methodological Conference (Kramatorsk, November 14-15, 2018) / under the general. ed. Dr. Tech. Science, Prof. SV Kovalevsky*. Kramatorsk: DDMA. P. 282-285. (in Russian)
- [9] Sedzhvik R. (2003) *Fundamental algorithms in C. Analysis. Data structures. Sorting. Search. Graph Algorithms*. SPb: OOO «DiaSoftJuP», 1136 p. (in Russian)
- [10] Mel'nikov A. Ju. (2013) *Object-oriented analysis and design of information systems: a tutorial*. Ed. 2nd, rev. and additional. Kramatorsk: DGMA, 172 p. (in Russian)

*Melnykov O.Yu., Sokolskyi O.S.*

### **SPECIAL APPLICATION OF OWN DEVELOPMENT FOR DEMONSTRATION AND COMPARISON OF ALGORITHMS FOR SORTING AND SEARCHING DATA**

**Abstract.** The article describes a special application of own design that allows students studying algorithms for sorting and searching data to observe the process and analyze the advantages and disadvantages of several methods to better understand the principles of their work. Some algorithms for sorting and searching data are considered, existing software systems (Internet sites) for solving the problem, their features, advantages, and disadvantages are analyzed. The development of an object-oriented model of the software system by means of visual modeling UML (diagrams of use cases and a class diagram are presented) and a functional model in BPWin notation (first and second levels are given). The available algorithms are listed: Bubble Sort, Insert Sort, Selection Sort, Merge Sort, Quick Sort, Shaker Sort, Gnome Sort, Shell Sort, Binary Sort, Sequential Search, Binary Search. Since the real operating time of the algorithm on a modern computer is too short, and the user will not have time to understand the principles of its operation, it was decided to add a delay after each step, which significantly increased the time when demonstrating the operation of one algorithm. Examples of using the developed application are given data entry and demonstration of the sorting algorithm by exchanges, comparison of sorting algorithms for randomly filling an array of 25000 elements according to the criteria "Running time" and "Number of iterations". The possibility of changing the interface language is noted. The use of the help system is described. A typical sequence of work with the created application is considered. It is concluded that the developed application can become an additional element of information and communication teaching aids in the presentation of relevant disciplines – for example, "Algorithms and data structures" for specialties of the industry 12 "Information technology".

**Keywords:** sorting and search algorithms, demonstration of work, complexity of the algorithm, UML modeling, functional model, software, application.

**DOI 10.31392/NPU-nc.series 2.2020.22(29).12**  
**УДК 378.016:373.5.011.3-051**

**Наталія Степанівна Павлова**

кандидат педагогічних наук, доцент,

доцент кафедри інформаційно-комунікаційних технологій та методики викладання інформатики

Рівненський державний гуманітарний університет, м. Рівне, Україна

ORCID ID 0000-0002-7817-6781

*nataliia.pavlova@rshu.edu.ua*

### **МЕТОДИЧНА ПІДГОТОВКА МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ ЯК СУЧАСНА ПЕДАГОГІЧНА ПРОБЛЕМА**

**Анотація.** Описано методичну підготовку майбутніх учителів інформатики в закладах вищої освіти як сучасну педагогічну проблему. Розкрито актуальність та доцільність удосконалення