Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
https://nv.nltu.edu.ua
https://doi.org/10.36930/40320511
Article received 16.10.2022 p.
Article accepted 27.10.2022 p.
UDC 004.(37+51)

ISSN 1994-7836 (print)
ISSN 2519-2477 (online)

@ Correspondence author
T. O. Korotyeyeva
tetyana.o.korotyeyeva@lpnu.ua

Д. Р. Козак, Т. О. Коротєєва
Національний університет "Львівська політехніка", м. Львів, Україна

ПРОЄКТУВАННЯ НАВЧАЛЬНОЇ СИСТЕМИ ВІЗУАЛІЗАЦІЇ РОБОТИ АЛГОРИТМІВ

Розглянуто особливості проектування навчальної системи для студентів спеціальності 121 "Інженерія програмного забезпечення" У межах дисципліни "Алгоритми і структури даних". Мета системи – полегшити процес вивчення основних принципів роботи алгоритмів через динамічну візуалізацію їх кроків та подання інформаційної складової (покроковий опис алгоритму та загальна інформація про нього) до кожного з них. Проведено дослідження візуалізації як методу навчання. Виявлено основні проблеми студентів під час вивчення теоретичного матеріалу. Проаналізовано сучасні засоби візуалізації покрокової роботи алгоритмів, які можна використовувати у навчальному процесі. Встановлено основні завдання системи і розроблено основні вимоги до програмного продукту. На основі виконаних досліджень спроєктовано архітектуру системи, яка є гнучкою до змін у сфері навчання. Побудовано структуру компонент програмного додатку, діаграму розгортання, діаграму прецедентів, діаграму класів і діаграму станів системи. Систему реалізовано як статичний веб-сайт для зручного способу доступу до ресурсів. Для реалізації системи використано React-бібліотеку для створення графічних інтерфейсів. Для графічної візуалізації використовувався примітив html canvas, що дає змогу зображати 2D графіку на веб-сторінці. Усі алгоритми в системі розбиті на категорії, а саме: сортування, пошуку стрічки, пошуку шляху та алгоритми на деревах. Для зручності кожна категорія представлена окремою веб-сторінкою. Загалом розглянуто 24 алгоритми. Користувач має змогу виконувати алгоритми покроково або автоматично, коли система сама здійснює наступний крок із певним заданим користувачем часовим інтервалом. Реалізована можливість виконання алгоритму у зворотному напрямку, Також користувач може згенерувати вхідні дані алгоритму або задати їх вручну. Розроблений веб-сайт є самодостатнім ресурсом для дистанційного вивчення дисципліни "Алгоритми і структури даних".

Ключові слова: архітектура системи; категорії алгоритмів; статичний веб-сайт; 2D графіка.

Вступ / Introduction

Два найцінніші ресурси для комп'ютерної програми – це час та пам'ять. Алгоритми і структури даних безпосередньо впливають на ці ресурси. Також існує відома формула "Програма = Алгоритми + Структури даних". Саме тому вивчення цієї теми є необхідною складовою частиною для формування фаху кожного програміста.

Саме ж завдання вивчення матеріалу тільки виглядає простою, насправді – це ціла наука. Потрібно, передусім, навчитися правильно споживати теорію. Доволі складно вивчити інформацію так, щоб не забути її через місяць чи рік [10]. Оскільки алгоритми та структури даних потрібні програмісту на щоденній основі, отже, і студент повинен вивчити цей матеріал так, щоб не забувати його роками.

Візуалізація навчального матеріалу полягає у використанні візуальних моделей у процесі навчання. Важливість візуалізації важко переоцінити, це одна з позитивних практик сучасного світу для вивчення теоретичного матеріалу [11, 12].

Отже, постає завдання провести проєктування та реалізацію навчальної системи візуалізації роботи алгоритмів.

Постановка завдання дослідження – створення навчальної системи візуалізації роботи алгоритмів для студентів, що вивчають дисципліну "Алгоритми і структури даних", яка дасть змогу краще опановувати теоретичні знання.

Об'єкт дослідження – проєктування системи для візуалізації роботи алгоритмів.

Предмет дослідження – методи і засоби проєктування навчальної системи для динамічної візуалізації роботи алгоритмів, що значно полегшить вивчення основних принципів і кроків виконання відповідних дій та подання інформаційної складової до кожного з них.

Мета роботи – спроєктувати та розробити навчальну систему для динамічної візуалізації роботи алгоритмів, яка забезпечить освоєння студентами основних принципів виконання відповідних кроків.

Для досягнення зазначеної мети визначено такі основні *завдання дослідження:*

- проаналізувати сучасні засоби візуалізації покрокової роботи алгоритмів, які можна використовувати у навчальному процесі для полегшення розуміння та освоєння студентами основних принципів виконання ним відповідних кроків;
- встановити основні завдання системи динамічної візуалізації роботи алгоритмів і розробити основні вимоги до програмного додатку;

Інформація про авторів:
Козак Дмитро Романович, бакалавр, кафедра програмного забезпечення. **Email:** dmytro.kozak.pz.2018@lpnu.ua
Коротєєва Тетяна Олександрівна, канд. техн. наук, доцент, кафедра програмного забезпечення.
Email: tetyana.o.korotyeyeva@lpnu.ua; https://orcid.org/0000-0002-6287-5895

Цитування за ДСТУ: Козак Д. Р., Коротєєва Т. О. Проєктування навчальної системи візуалізації роботи алгоритмів. Науковий вісник НЛТУ України. 2022, т. 32, № 5. С. 80–86.
Citation APA: Kozak, D. R., & Korotyeyeva, T. O. (2022). Design of an educational system for visualizing the work of algorithms. *Scientific Bulletin of UNFU*, 32(5), 80–86. https://doi.org/10.36930/40320511

- визначити структуру компонент програмного додатку, побудувати його архітектуру та провести проєктування інтерфейсу користувача;
- визначити візуальну й інформаційну складові для кожного алгоритму;
- розробити веб-систему для динамічної візуалізації роботи алгоритмів, яка значно полегшить розуміння та освоєння студентами основних принципів виконання відповідних кроків.

Аналіз останніх досліджень та публікацій. Існує низка наукових робіт, які базуються на конкретних дослідженнях над здатністю людей запам'ятовувати інформацію. І всі вони приходять до висновку, що перечитування інформації не покращує розуміння матеріалу [10, 12]. Хорошими методами навчання є отримання інформації з пам'яті (active recall) та інтервальне повторення (spaced repetition). Прикладом отримання інформації з пам'яті є запитання наприкінці навчального матеріалу, вони змушують мозок працювати й запам'ятовувати інформацію глибше. Метод інтервального повторення полягає у повторенні інформації через певний проміжок часу, який постійно збільшується. Це дасть змогу інформації міцно закріпитися у вашій голові. Проте це, звісно ж, важче, ніж просто перечитати матеріал.

Сучасний студент має певні особливості мислення, які визначають ефективність візуалізації: здатність до швидкого оброблення інформації та до швидкого переключення уваги; непристосованість сприйняття великої лінійної інформації (книжки); добре сприйняття інформації, яку подано графічно [10, 11]. Якщо перевести навчальну інформацію з різних каналів сприйняття і зобразити це у візуальній формі, то отримаємо пришвидшення оброблення й засвоєння матеріалу. Використання візуалізації заощаджує час вивчення матеріалу, адже така інформація є більш чіткою, лаконічною та доступною. У процесі розуміння цих даних студенти навчаються аналізувати отриману інформацію, зіставляти її з вивченою, знаходити слабкі місця в їхньому розумінні матеріалу, висловлювати власну думку, бачити залежності.

Дослідження цього і висновки потреби візуалізації подано в роботі [8]. Візуалізацію розглядають як потужний чинник активації різних видів пам'яті та мислення людини.

У такий спосіб можна сформулювати основні переваги візуалізації:

- полегшує розуміння навчальної інформації, подаючи її відповідно до когнітивних особливостей сучасних студентів;
- сприяє формуванню достовірних уявлень про об'єкт, що вивчається;
- дає можливість студентам концентрувати увагу на головному матеріалі навчального процесу і відсікати другорядні деталі;
- дає змогу пришвидшити процес вивчення матеріалу;
- надає легкий і приємний спосіб повторення матеріалу;
- активізує різні види пам'яті та мислення;
- розвиває уяву;
- створює позитивний фон навчання;
- полегшує комунікацію студентів між собою на тему об'єкта, що вивчається.

Аналіз сучасних програмних засобів-аналогів проєктованої системи. Великий попит породжує велику кількість продуктів, тому систем-аналогів для навчання є доволі багато. Написання такої системи візуалізації – це також і хороша вправа на тренування знань алгоритмів, тому люди часто створюють її без комерційної вигоди, заради кращого розуміння, і викладають у вільний доступ. Такі програми не задовольняють усіх потреб студентів, адже їх розробляли не для них та не враховували різноманітні особливості процесу навчання.

У мережі Інтернет існує багато різноманітних вебсайтів візуалізації роботи різних алгоритмів. Ось декілька з них: algorithm-visualizer.org [14]; clementmihailescu.github.io/Pathfinding-Visualizer [15].

На рис. 1 наведено приклад візуалізації алгоритму сортування із ресурсу [14]. На рис. 2 зображено приклад візуалізації алгоритму пошуку шляху [15].

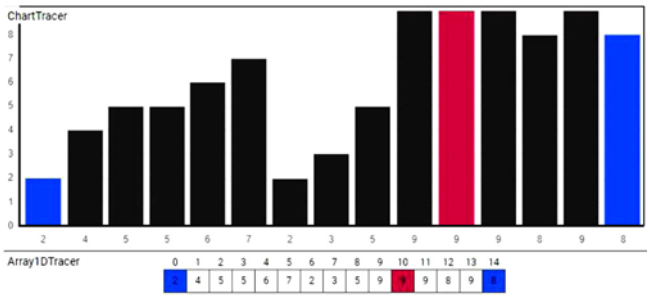

Рис. 1. Приклад алгоритму сортування / An example of a sorting algorithm

Рис. 2. Приклад алгоритму пошуку шляху / An example of a path search algorithm

Усі ці аналоги доволі схожі. Вони візуалізують алгоритм по-різному, адже немає ніяких точних правил, які б говорили як потрібно візуалізувати ту чи іншу річ. Переглядаючи системи-аналоги, можна отримати уявлення про різні можливості візуалізації й обрати той, який найбільше підходить. Але всі вони орієнтовані на візуалізацію тільки певного алгоритму або певної категорії алгоритмів та не дають змогу охопити весь теоретичний матеріал дисципліни "Алгоритми і структури даних".

Серед платних ресурсів було розглянуто систему algoexpert [13] – комерційний приклад навчальної системи, що орієнтована на алгоритми. У ній більше уваги приділено теоретичній складовій для підготовки до технічних інтерв'ю. Аналіз algoexpert дав змогу визначити необхідні складові для майбутньої системи, відмінної від наявних, що забезпечить зв'язок теоретичного і практичного матеріалу через їх детальний опис та демонстрацію отриманого результату.

Результати дослідження та їх обговорення / Research results and their discussion

Опис дослідження. Завдання полягає у створенні веб-сайту, який буде відображати покрокове виконання алгоритмів, що вивчаються в межах дисципліни "Алгоритми і структури даних". Окрім частини візуалізації, система буде містити інформацію про кожен із алгоритмів, а також його покроковий опис, який повинен бути описаний за стилістикою, що описана в [4].

Алгоритми, які представлені в системі, поділятимуться на 4 категорії:

- сортування – впорядкування лінійних даних у висхідному порядку;
- пошук стрічки – пошук входження зразка стрічки в тексті;
- пошук шляху – пошук маршруту від початкової точки до кінцевої на дискретному полі;
- алгоритми на деревах – сортування та обходи дерев.

Існує тільки один клас користувачів, який має доступ до усіх можливостей системи. Він повинен мати змогу налаштувати вхідні дані: згенерувати їх чи задати вручну.

Основними можливостями системи є:

- візуалізація чотирьох категорій алгоритмів: сортування, пошук зразка тексту, пошук шляху, алгоритми на деревах;
- можливість виконувати алгоритм покроково та автоматично;
- можливість задавати часовий інтервал виконання кожного кроку алгоритму;
- можливість зупиняти роботу алгоритму;
- переглянути коротку інформацію та покроковий опис про кожен з алгоритмів;
- можливість змінювати вхідні дані алгоритмів;
- чітке і зрозуміле колірне відображення даних на поточному кроці алгоритму.

Основні засоби виконання завдання. Система складається тільки з клієнтської частини веб-сайту. Основною бібліотекою для виконання цього буде React [17] – бібліотека для створення графічних інтерфейсів на основі JavaScript. Доцільність цієї технології полягає в її легкості й популярності, а також у використанні декларативного способу написання інтерфейсу. Перевагою React є можливість організації частин інтерфейсу в окремі компоненти, які можна використовувати в різних місцях без дублювання коду. Також React дає змогу задати шляхи для різних сторінок. Сам React дає змогу створити односторінковий веб-сайт, тобто весь контент з http серверу буде надходити одразу, а дії користувача не будуть призводити до повторного запиту отримання нової сторінки.

Для графічної візуалізації буде використовуватися примітив html canvas, що дає можливість зображати 2D графіку на веб-сторінці. JavaScript дає змогу обробляти події користувача стосовно canvas, тому можна реагувати на кліки, рухи мишкою. Також JavaScript надає інтерфейс доступу до полотна.

Розгортання системи буде відбуватися з використанням GitHub Pages, для безоплатного і постійного хостингу веб-сайту. Очевидна перевага – не потрібно оплачувати існування веб-серверу та гарантія постійної доступності веб-сайту.

Проєктування архітектури системи. Важливим завданням розробки архітектури є створення компонентів системи, які б давали змогу легко змінювати поточні алгоритми або створювати нові. Тому алгоритми є окремою частиною системи. Алгоритми не мають нічого знати про інтерфейс системи, проте вони повинні мати певні функції, які давали б змогу змінювати поточний стан даних, над якими застосовується алгоритм (наприклад, змінити колір відображення даних). Модуль візуалізації даних повинен знати їхній стан на кожному кроці виконання, тобто мати прямий доступ до даних, які алгоритм опрацьовує. Компонента UI повинна вміти виконувати алгоритми при кожній зміні вхідних даних. Така архітектура нагадує шаблон проєктування MVC (англ. Model, View, Controller) [3]. На рис. 3 подано діаграму розгортання, на якій присутні описані вище компоненти системи.

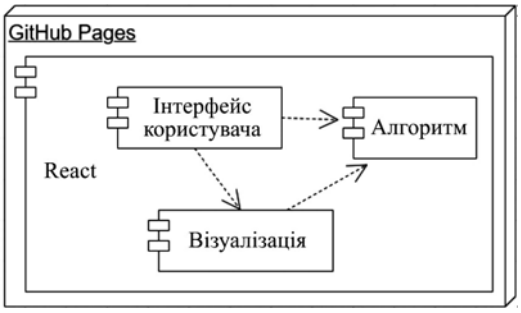

Рис. 3. Діаграма розгортання системи візуалізації алгоритмів / Algorithm visualization system deployment diagram

Проєктування поведінки системи. Оскільки в цій системі тільки один клас користувача, він матиме змогу виконувати всі функції, що надає система. Основний перелік функцій системи зображено на діаграмі прецедентів (рис. 4). Актор-користувач, яким для системи є студент, що вивчає алгоритми, може переходити по сторінках категорій алгоритмів, обирати потрібний алгоритм, генерувати дані, визначати часовий інтервал для виконання кроків алгоритму або задавати автоматичне виконання, проводити запуск роботи алгоритму та візуально спостерігати за його кроками виконання. Також користувач має змогу прочитати теоретичну інформацію про цей алгоритм, яку подано у покроковому описі.

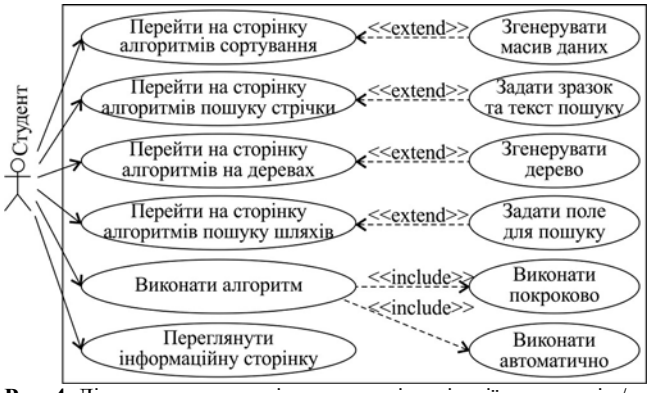

Рис. 4. Діаграма прецедентів системи візуалізації алгоритмів / Algorithm visualization system use case diagram

Окрема сторінка категорії алгоритмів повинна реалізовувати однаковий інтерфейс виконання алгоритмів, який можна описати діаграмою станів (рис. 5).

Кожен алгоритм повинен виступати окремим класом, який використовує певні надані інтерфейси для опрацювання структур даних. Таким інтерфейсом буде виступати базовий клас алгоритму. Кожна категорія алгоритмів повинна містити базовий клас, від якого успадковуються всі алгоритми (рис. 6). Це дає змогу легко добавити новий алгоритм, який буде імплементувати аналогічний інтерфейс доступу до даних. Така реалізація повинна нагадувати шаблон проєктування "Фабричний метод" [3]. Алгоритми повинні реалізовуватися не за допомогою модифікації вхідних даних, а за допомогою збереження кожного його окремого кроку в масив, який буде містити всю інформацію для його відображення. Модифікація даних відбувається на вимогу користувача. Алгоритм виконується ще задовго до того, як користувач вирішить його візуалізувати, а саме, під час задавання чи зміни вхідних даних. Таке рішення зумовлене наявністю покрокового виконання, яке неможливе, якщо не виконати алгоритм наперед і не помісти-

ти результат кожного кроку в певну окрему структуру даних.

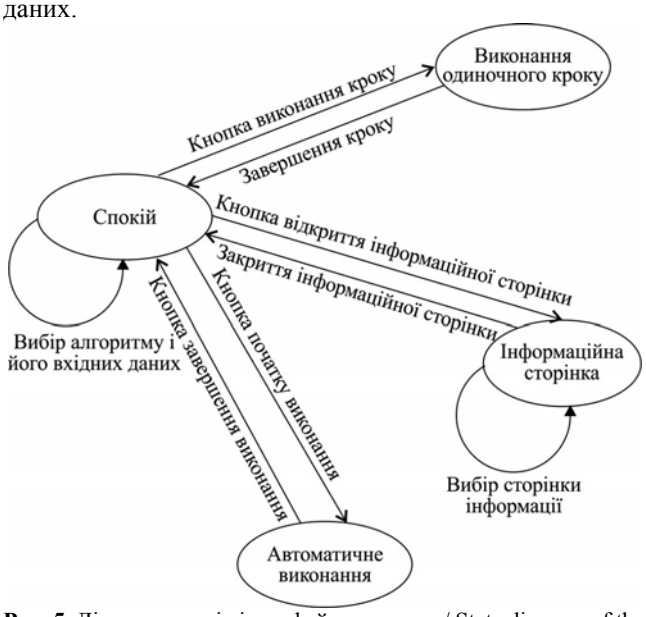

Рис. 5. Діаграма станів інтерфейсу системи / State diagram of the system interface

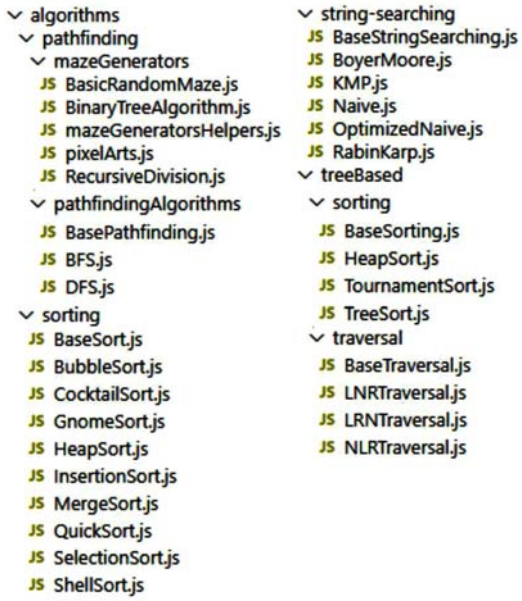

Рис. 6. Перелік класів алгоритмів / List of algorithm classes

Візуалізація відбувається за допомогою примітиву html canvas. Для відображення стану даних потрібна додаткова інформація, яка б говорила про їх поточний стан. Кожен елемент даних (вузол у деревах, клітинка поля в пошуку шляху, елемент масиву в сортуванні, символ у пошуку тексту) буде містити в собі інший об'єкт, що вказує на стан і вже знає, як потрібно зобразити цей конкретний елемент (якого кольору він повинен бути, чи повинна бути анімація, чи зображати певний текст).

Для кожного стану елемента створено окремий клас (стан спокою, стан порівняння елемента, стан зміни елемента, стан появи елемента і т.ін.) (рис. 7). Кожен елемент повинен мати інтерфейс зміни типу (стану), щоб алгоритм, який опрацьовує дані, міг задати цей тип. А сама сторінка повинна мати доступ до функції малювання, яка реалізується кожним класом стану, щоб знати, як зобразити елемент на сторінці. Такий опис елементів схожий до опису поведінкового патерну проєктування "Стан" [3].

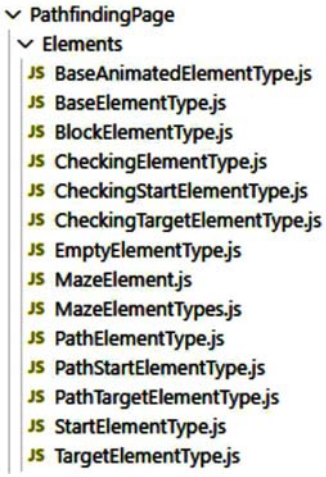

Рис. 7. Перелік станів елементу пошуку шляху / List of the pathfinding element states

Окрема сторінка категорії алгоритмів реалізує власну функцію малювання, яка викликається кожні 20 мілісекунд, що дає змогу отримати приблизно 50 кадрів за секунду. Описану вище архітектуру зображено на діаграмі класів для категорій алгоритмів (рис. 8).

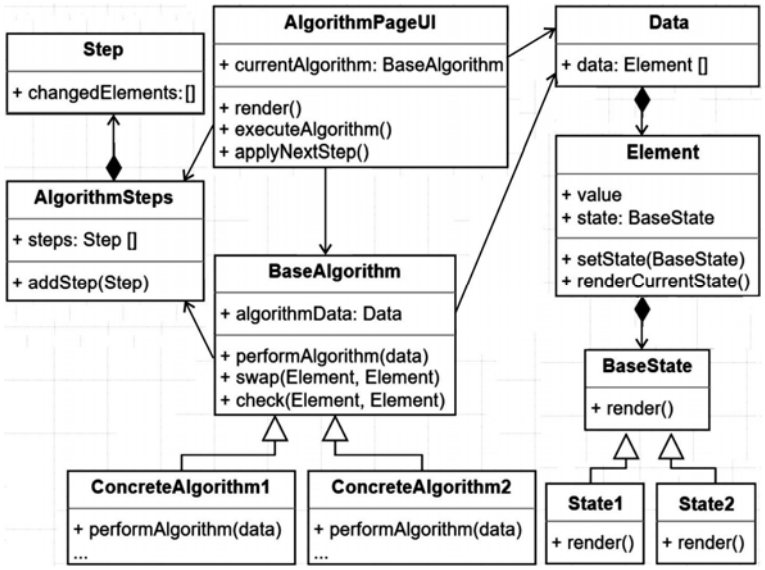

Рис. 8. Діаграма класів системи візуалізації алгоритмів / Algorithm visualization system class diagram

Проєктування інтерфейсу користувача. Головна сторінка (рис. 9) розробленої системи містить посилання на сторінки виконання алгоритмів та складається з чотирьох категорій, а саме, алгоритми сортування, алгоритми пошуку стрічки, алгоритми на деревах та алгоритми пошуку шляху.

Рис. 9. Головна сторінка веб-сайту / Website home page

Сторінка алгоритмів сортування (рис. 10) на конфігураційній панелі містить такі елементи: меню вибору алгоритму, слайдер задавання розміру масиву, слайдер задавання тривалості кроку автоматичного виконання, панель виконання алгоритму та іконку інформаційної сторінки. Основне поле сторінки наповнене прямокутниками голубого кольору, які відображають розміщення елементів масиву та їх значення (висота прямокутника). У момент виконання чергового кроку алгоритму прямокутники, які порівнюються за значенням і змінюють свої позиції, зафарбовуються в жовтий колір. Панель виконання алгоритму містить чотири кнопки: покрокове та автоматичне виконання алгоритму в обох напрямках. У системі реалізовано 9 різних алгоритмів сортування, покроковий опис яких можна прочитати в інформаційній сторінці.

Сторінка алгоритмів пошуку стрічки (рис. 11) на конфігураційній панелі містить такі елементи: меню вибору алгоритму, кнопку завершення виконання пошуку (вона необхідна, щоб повернути можливість введення тексту, яка зникає після початку виконання алгоритму), слайдер задавання тривалості кроку автоматичного виконання, панель виконання алгоритму та іконку інформаційної сторінки. Дві стрічки (слово для пошуку і текст, в якому відбувається пошук) розміщують одну під одною і вирівнюють по слову. Жовтим кольором позначені символи тексту і слова в момент порівняння. У разі успішного завершення роботи алгоритму, знайдене слово у тексті зафарбовується зеленим кольором. Якщо слово не знайдене у тексті, воно зафарбовується червоним кольором. У системі реалізовано 5 різних алгоритмів пошуку стрічки.

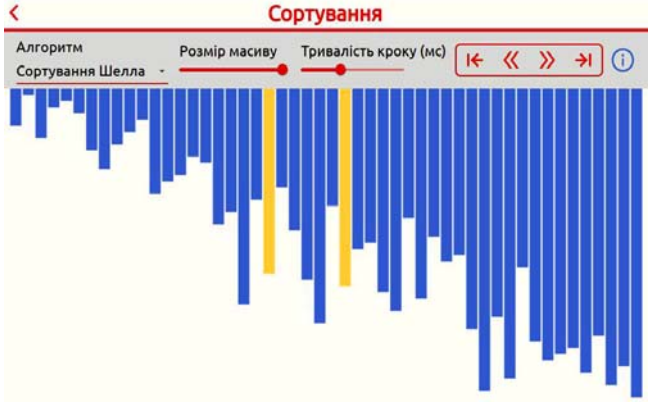

Рис. 10. Сторінка алгоритмів сортування / Sorting algorithms page

Рис. 11. Сторінка алгоритмів пошуку стрічки / String-searching algorithms page

За цим самим принципом сторінка алгоритмів на деревах (рис. 12) містить меню вибору алгоритму, слайдер задавання розміру дерева, слайдер задавання тривалості кроку автоматичного виконання, панель виконання алгоритму та іконку інформаційної сторінки. Кожен вузол дерева містить у собі певне числове значення. Користувач може генерувати випадкове дерево, але завжди таке, яке поміщається на екрані. Голубим кольором позначені вхідні вузли дерева. Зеленим кольором позначені вузли дерева, які при виконанні алгоритму займають остаточну позицію. Червоним кольором фіксуються активні вузли на момент виконання певного кроку алгоритму. В системі реалізовано 3 алгоритми обходу дерева та 3 алгоритми сортування на деревах.

Аналогічно організована сторінка алгоритмів пошуку шляху (рис. 13). Поле лабіринту розбите на комірки та передбачає присутність перешкод (комірки коричневого кольору). Користувач може власноруч будувати перешкоди та переміщати початкову і кінцеву точки шляху. У кожної комірки є 4 сусіди, до яких прокладено шлях, якщо його не перекрито перешкодою. Процес побудови варіантів шляху позначено рожевим кольором, а остаточний побудований шлях – червоним кольором. В системі реалізовано 4 алгоритми пошуку шляху.

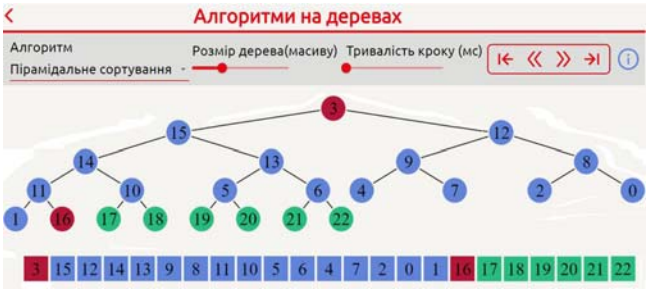

Рис. 12. Сторінка алгоритмів на деревах / Tree-based algorithms page

Рис. 13. Сторінка алгоритмів пошуку шляху / Pathfinding algorithms page

Інформаційна сторінка (рис. 14) містить меню вибору інформації, де можна обрати або загальну інформацію про категорію алгоритмів, або інформацію про певний алгоритм, який реалізований у системі. Інформація про такий алгоритм включає його покроковий опис та інформацію про нього (складність, опис, порівняння з

іншими алгоритмами). Кожна категорія алгоритмів містить пункт "Загальна інформація", де можна прочитати інструкцію для роботи з цією сторінкою.

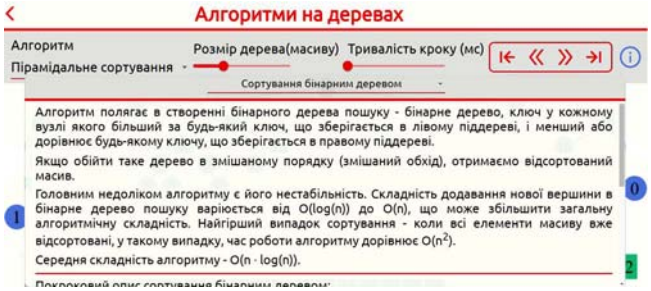

Рис. 14. Інформаційна сторінка / Information page

Обговорення результатів дослідження. Проведене дослідження дало змогу отримати такі основні результати: спроєктовано та реалізовано систему візуалізації роботи алгоритмів, що вивчаються в межах дисципліни "Алгоритми і структури даних". Система реалізована як статичний веб-сайт на основі бібліотеки для створення графічних інтерфейсів. Для графічної візуалізації використано примітив html canvas.

Важливими ресурсами у проєктуванні системи стали algoexpert [13] як еталон теоретичної інформації, яка подана найкраще для студентів, та [14, 15], як хороші приклади візуалізації алгоритмів. Також прикладом електронної системи навчання стала робота [7], в якій описано основні вимоги, які можуть бути застосовані для вивчення алгоритмів. Окрім цього, було проаналізовано роботи, які акцентують увагу на візуалізації даних [6, 5].

Спроєктована навчальна система візуалізації роботи алгоритмів є доцільним поєднанням досліджених ресурсів, що дає змогу студентам залишатися в одному середовищі для вивчення всієї дисципліни "Алгоритми і структури даних".

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – розроблено навчальну систему для динамічної візуалізації роботи алгоритмів, яка, на відміну від наявних, забезпечує зв'язок теоретичного і практичного матеріалу через їх детальний опис та демонстрацію отриманого результату.

Практична значущість результатів дослідження – спроєктовану та програмно реалізовану навчальну систему для динамічної візуалізації роботи алгоритмів можна використовувати у навчальних закладах для практичного вивчення теоретичного матеріалу з дисципліни "Алгоритми і структури даних".

Висновки / Conclusions

Розроблено навчальну систему для студентів спеціальності 121 "Інженерія програмного забезпечення" в межах дисципліни "Алгоритми і структури даних", яка дає змогу полегшити процес вивчення основних принципів роботи алгоритмів через динамічну візуалізацію їх кроків та подання інформаційної складової до кожного з них. За результатами проведеного дослідження можна зробити такі основні висновки.

1. Проаналізовано останні дослідження та публікації, які дали змогу сформувати сутність об'єкта та предмета дослідження. Проведено дослідження візуалізації як методу навчання. Виявлено основні проблеми студентів під час вивчення теоретичного матеріалу.
2. Встановлено основні завдання системи і розроблено вимоги до програмного продукту. На основі проведених досліджень спроєктовано архітектуру системи, побудовано структуру компонент програмного додатку, діаграму розгортання, діаграму прецедентів, діаграму класів та діаграму станів системи.
3. Систему реалізовано як статичний веб-сайт для зручного способу доступу до ресурсів. Для реалізації системи використано React-бібліотеку для створення графічних інтерфейсів. Для графічної візуалізації використовувався примітив html canvas.
4. Навчальну систему для динамічної візуалізації роботи алгоритмів можна використовувати у закладах освіти для практичного вивчення теоретичного матеріалу з дисципліни "Алгоритми і структури даних".

Завдяки засобам динамічної візуалізації система підвищує інтерес студентів до вивчення теоретичного матеріалу дисципліни "Алгоритми і структури даних". Серед її переваг можна виділити такі: полегшує розуміння навчальної інформації, подаючи її відповідно до когнітивних особливостей сучасних студентів; сприяє формуванню достовірних уявлень про об'єкт, що вивчається; полегшує комунікацію студентів між собою; створює позитивний фон навчання.

References

1. Auden, E., Toutain, T., & Zharkov, S. (2007). eSDO algorithms, data centre and visualization tools. *Astronomical Notes, 328*(3-4), 356–361. https://doi.org/10.1002/asna.200610742
2. Carlos, A. S., Oliveira, Panos, M. Pardalos, & Prokopyev, O. A. (2007). Data Structures and Algorithms. *Wiley Encyclopedia of Computer Science and Engineering*. https://doi.org/10.1002/9780470050118.ecse097
3. Craig, Larman, & Kent, Beck. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edition. Great Britain: Pearson, 736. [In English].
4. Donald, Knuth. (2021). Art of Computer Programming, Volume 3: Sorting and Searching. Kyiv: Dialektyka, 832. [In Ukrainian].
5. Evergreen, S., & Metzner, C. (2013). Design Principles for Data Visualization in Evaluation. New Dir. Eval. 2013(140), 5–20. https://doi.org/10.1002/ev.20071
6. Frits, H. Post. (2011). Data Visualization: Featuring Interactive Visual Analysis. *Computer Graphics Forum, 30*(2). https://doi.org/10.1111/j.1467-8659.2011.01911.x
7. Kukharenko, V. M. (2020). Development of a modern e-learning system in the university. *Ukrainian Journal of Information Technologies*, 2(1), 95–102. https://doi.org/10.23939/ujit2020.02.095
8. Linsiya, Patrao. (2019). Needs and Benefits of Data Visualization. Edureka! Ebook (New). Data Visualization with Tableau. Retrieved from: https://www.edureka.co/blog/needs-and-benefits-of-data-visualization/
9. Miles, R., & Hamilton, K. (2006). Learning UML 2.0: A Pragmatic Introduction to UML. California: OReilly Media, 290. [In English].
10. Peter, Brown, Henry, Roediger, & Mark, McDaniel. (2019). Make it stick. The Science of Successful Learning. Kyiv: Nash Format, 240. [In Ukrainian].
11. Scott, Berinato. (2022). Good Charts Workbook: Tips, Tools, and Exercises for Making Better Data Visualizations. Kyiv: ArtHuss, 288. [In Ukrainian].
12. Shakti, Gawain. (2010). Creative Visualization: Use the Power of Your Imagination to Create What You Want in Your Life. California: New World Library, 192. [In English].
13. Website. (2022). Algoexpert.io. Streamlined platform for learning algorithms. Retrieved from: https://www.algoexpert.io/product. [In English].

14. Website. (2022). An example of visualization of algorithms. Retrieved from: https://algorithm-visualizer.org. [In English].
15. Website. (2022). An example of visualization of algorithms. Retrieved from: https://clementmihailescu.github.io/Pathfinding-Visualizer/. [In English].
16. William, Lidwell, Kritina, Holden, & Jill, Butler. (2010). Universal Principles of Design: 125 Ways to Enhance Usability. Cummings Center: Rockport Publishers, 280. [In English].
17. Zac, Gordon, Mikall, Angela, Hill, & Robbie, Adair. (2019). React Explained: Your Step-by-Step Guide to React. Independently published, 366. [In English].

D. R. Kozak, T. O. Korotyeyeva

Lviv Polytechnic National University, Lviv, Ukraine

DESIGN OF AN EDUCATIONAL SYSTEM FOR VISUALIZING THE WORK OF ALGORITHMS

The purpose of this work was to create software for visualization of algorithms studied in the course "Algorithms and data structures". In the course of research, we developed a website where students can get additional information while studying the discipline, in particular visually see how a certain algorithm is executed and read the description of this algorithm. This will facilitate the process of learning the basic principles of algorithms and allow students to master the discipline better. The educational system will not only improve students' knowledge, but also enhance university teaching of the discipline. The subject area is studied and the main advantages of visualization as a teaching method are described. The main problems of students when studying the educational material are examined. Modern programs that can be used in the educational process to visualize the step-by-step execution of algorithms were also analyzed. On the basis of the conducted research, we developed system architecture, which is flexible to changes in the field of education. A requirements specification has been created. To better understand the system, the following diagrams were created: use case diagram, state diagram, deployment diagram, class diagram. The website was developed using React – a library for creating graphical user interfaces. The html canvas primitive was used for visualization rendering, which allows easy displaying of 2D graphics on the web page. The deployment took place using GitHub Pages. The obvious advantage is that there is no need to pay for the existence of a web server and a guarantee of constant availability of the website. The system implements algorithms of the following four categories: sorting, string-searching, pathfinding, and tree-based algorithms. Separate pages have been created for each of the categories. An information page has also been created, which can be opened from any algorithm category page and will contain information about these algorithms. Similar products were investigated in order to provide good visualization. The main practices were clarified and applied in our system. Visualizations of 24 algorithms are implemented, each of which clearly shows how the algorithm works. Users can perform the algorithms step by step, where they can decide when the next step will take place, or automatically, where the system itself performs the next step with a certain interval specified by the user. It is possible to perform the algorithm in reverse, so you can return to the previous step. The user can also generate the input data of the algorithm, or specify them manually. Thus, we can draw the conclusion that all aspects investigated in this work prove that the algorithm visualization system is really useful during studying. The developed system can be easily scaled, i.e., it can be expanded with new algorithms when the curriculum changes.

Keywords: system architecture; algorithm categories; static website; 2D graphics.