

УДК 378.14

DOI: 10.30977/BUL.2219-5548.2022.96.0.146

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИВЧЕННЯ ПРОГРАМУВАННЯ ТА АЛГОРИТМІЗАЦІЇ З ДОПОМОГОЮ ВИКОРИСТАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ТЕСТУВАННЯ

Щербаков О. В., Скорін Ю. І.

Харківський національний економічний університет ім. С. Кузнеця

***Анотація.** У статті розглядаються шляхи підвищення ефективності вивчення основ програмування та алгоритмізації. Перспективною формою підвищення ефективності навчального процесу є використання автоматизованих систем тестування. Аналіз наявних підходів до вирішення цього питання дозволив виокремити декілька різних варіантів організації навчального процесу з використанням автоматизованих систем тестування та виявити переваги й недоліки кожного з них. За результатами досліджень було запропоновано застосування автоматизованої тестувальної системи, що дозволяє швидко та ефективно організувати процес автоматичної перевірки програм студентів, надає студентам змогу в будь-який час перевірити свої рішення, а також залишає викладачеві можливість максимально контролювати навчальний процес і керувати ним.*

***Ключові слова:** програмування, алгоритмізація, автоматизована система тестування, ефективність навчального процесу.*

Вступ

Ефективне та якісне вивчення студентами ІТ-спеціальностей основ програмування та алгоритмізації є запорукою підготовки висококваліфікованих фахівців для ІТ-галузі, вимоги до рівня підготовки яких щороку зростають. Поява нових мов програмування, розвиток теоретичних основ комп'ютерних технологій вимагають постійного вдосконалення змісту навчальних дисциплін і навчальних планів. Аналіз міжнародних стандартів у галузі комп'ютерних наук [1] свідчить про необхідність формування у випускників ІТ-спеціальностей університетів широкого спектру професійних та фахових компетентностей у сфері інформаційних технологій, зокрема здатність алгоритмічного мислення, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їхньої ефективності та складності, здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування та здатність формулювати й забезпечувати вимоги щодо якості програмного забезпечення. Одними з основних фундаментальних дисциплін у підготовці майбутніх ІТ-фахівців є дисципліни «Програмування» та «Алгоритми та структури даних». Їхньою метою є вивчення основних алгоритмічних і синтаксичних конструкцій мов програмування, ефективних структур і алгоритмів оброблення даних в інформацій-

них системах, формування у студентів компетентностей у проектуванні, розробленні й аналізі алгоритмів, оцінюванні їхньої складності для адекватного моделювання предметних галузей, а також реалізації алгоритмів у вигляді програмних та інформаційних систем. У більшості університетів у межах дисципліни «Алгоритми та структури даних» вивчаються такі теми, як алгоритми сортування та пошуку, алгоритми на графах, алгоритми обчислювальної геометрії, алгоритми пошуку в рядках, комбінаторні й евристичні алгоритми. Основними структурами даних, що розглядаються в межах цієї дисципліни, є лінійні списки, черги, стеки, дерева та графи [2–4]. Реалізація всіх перелічених вище алгоритмів і структур даних у вигляді програм, з одного боку, дозволяє студентам глибше зрозуміти теоретичний матеріал і набути практичних навичок його використання, а з іншого – дає викладачам змогу переконатися, що в студентів сформувалися необхідні компетентності. Але водночас актуальним стає питання тестування програм, що розробляють студенти. Очевидно, що так зване ручне тестування є надто трудомістким процесом і малоефективним, тому все більше викладачів використовують автоматизовані тестувальні системи.

Аналіз публікацій

Проведений аналіз публікацій [5–8] показав, що використання в навчальному процесі

автоматизованих систем тестування є досить поширеним підходом. У багатьох навчальних закладах для вивчення дисциплін, пов'язаних з алгоритмізацією та програмуванням, використовуються різні автоматизовані засоби навчання та перевірки. Можна виокремити два основні підходи до організації автоматизованого тестування студентських програм.

Перший підхід полягає у використанні спеціальних інформаційних систем, що виконують автоматичну перевірку програм. Одне з визначень поняття «автоматизована система тестування» наведено в [7] і сформульовано так: це комплекс програмних (або програмно-апаратних) засобів, які дозволяють в автоматичному режимі проводити тестування рішень користувачів у межах об'єкта тестування. Існує досить значна кількість таких систем, наприклад, такі як PCMS2, Contester, DOMjudge тощо, але однією з найпоширеніших є Ejudge [9]. Ejudge – це система для проведення різноманітних заходів, в яких потрібна автоматична перевірка програм. Незважаючи на те, що основне призначення системи Ejudge – це проведення олімпіад та інших змагань із програмування, система знайшла також широке застосування в навчанні програмування в різних університетах [8]. Але такий підхід має один істотний недолік. Система Ejudge не має бази завдань із тестами, викладач повинен їх розробити самостійно. Ця обставина суттєво ускладнює процес і потребує чимало часу.

Другим можливим варіантом застосування автоматизованих систем тестування для вивчення алгоритмів і структур даних, позбавлених зазначеного недоліку, є використання інтернет-ресурсів, розрахованих на застосування в процесі навчання або тренування вже готових завдань. Типовими представниками таких систем є сайт «Школа програміста» [10] або сайт «E-olymp» [11]. База даних кожного з цих двох сайтів містить понад тисячу завдань різного рівня складності. Методика використання цих ресурсів для підготовки учнів до олімпіад із програмування викладена в роботах [12–14]. Наприклад, на сайті «E-olymp» викладач може створювати групи учнів (студентів), запрошувати до цих груп учасників, які попередньо вже зареєструвалися на цьому сайті, створювати на базі наявних завдань різноманітні змагання, переглядати загальний рейтинг учасників (за результатами всіх змагань) та рейтинг за результатами окремих змагань. Хоча основним призначенням сайту є підготовка до

олімпіад, про що говорить сама назва сайту, його можна використовувати й на заняттях із програмування [14].

Безумовно, наявність готової бази завдань суттєво скорочує час на підготовку турніру. Але разом із тим відсутність можливості побачити код кожного рішення студента, відсутність можливості анулювати рішення, якщо його автор порушує встановлені викладачем вимоги, інші обмеження знижують ефективність такого підходу саме в процесі вивчення алгоритмізації.

Мета та постановка завдання

Проведений аналіз довів перевагу використання автоматизованих систем тестування для перевірки програм студентів перед традиційними підходами, так званим «ручним тестуванням».

Разом із тим, не всі системи автоматизованого тестування можна легко адаптувати для проведення навчального процесу. Більшість систем, які аналізувалися, мають певні недоліки, що знижують ефективність їхнього застосування в цьому питанні.

Метою цієї статті є аналіз наявних підходів і пошук ефективного вирішення проблеми автоматизованого тестування програм студентів під час вивчення програмування та алгоритмізації.

Виклад основного матеріалу

Автори статті вже кілька років під час проведення лабораторних і практичних занять із навчальних дисциплін «Програмування», «Основи алгоритмізації», «Алгоритми та структури даних» використовують систему DOTS. DOTS (Distributed Olympiad Test System) – це онлайн-платформа для навчання програмування й розвитку алгоритмічного мислення [15]. Система дозволяє студентам більш ефективно вивчати алгоритмізацію та програмування, дає змогу в будь-який момент відправити своє рішення у вигляді програми на перевірку та практично миттєво отримати вердикт перевіркою системи на своє рішення.

Широкі можливості система надає викладачам. Він може для кожної дисципліни для кожної групи створити лабораторну або практичну роботу (у системі вони називаються «турніри»), наповнити її завданнями з бази готових завдань, установити час початку та закінчення й отримує максимальний контроль над процесом виконання студентами завдань. Система підтримує 28 різних компіляторів усіх поширених мов програмування, таких як

C, C++, Java, C#, Python, Kotlin, Haskell, Ruby, JavaScript тощо.

Система має значну базу готових завдань, поділених на три великі групи – завдання для новачків на елементарні алгоритмічні конструкції, такі як розгалуження або цикли, завдання для вивчення алгоритмів і структур даних, а також завдання олімпіадного характеру. У межах кожної групи всі завдання також поділені на теми. Наприклад, для вивчення основ програмування передбачені окремо задачі на лінійні алгоритми, задачі, для реалізації яких потрібно використовувати умовні оператори й задачі на циклічні алгоритми. А для вивчення алгоритмів та структур даних виокремлено такі теми: алгоритми цілочисельної арифметики, алгоритми сортування, алгоритми на графах, алгоритми на деревах, завдання на динамічне програмування, алгоритми роботи з рядками, алгоритми обчислювальної геометрії, комбінаторні алгоритми тощо. Кожне завдання має свій рівень складності, і викладач легко та швидко може сформувати комплект завдань для кожної навчальної групи й навіть для окремих студентів з урахуванням рівня попередньої алгоритмічної підготовки. За необхідності викладач може будь-якої миті змінити набір завдань, додавши до нього або додаткові завдання, або, навпаки, замінити складне завдання на більш легке.

Усі завдання з алгоритмів та структур даних у системі мають однаковий формат. Існує чітке формулювання завдання, указаний формат вхідних даних, в якому прописані всі обмеження на вхідні дані, а також вказано порядок введення вихідних даних. Обов'язково вказується формат вихідних даних, щоб студент розумів, у якій послідовності виводити результати. Це дуже важливий момент, оскільки за умови автоматичної перевірки система порівнює результат роботи програми студента з результатом еталонного рішення й неправильний формат виведення результатів може призвести до вердикту WA – Wrong Answer (неправильна відповідь). Крім того, обов'язково вказується обмеження на час виконання програми й обмеження на кількість пам'яті, що виділяється. Отже, автоматично перевіряється, наскільки ефективним у плані обчислювальної складності є рішення студента. Якщо рішення виконується довше за встановлений час, система автоматично припиняє подальше тестування та видає вердикт TL – Time Limit Exceeded (перевищено час виконання). Якщо ж рішення вимагає більше

пам'яті, ніж передбачено обмеженнями завдання, система видає повідомлення ML – Memory Limit Exceeded (перевищено обмеження пам'яті).

Для кожного турніру система автоматично формує рейтингову таблицю, де вказується кількість балів, набраних кожним студентом групи за кожне завдання, загальна кількість балів, набрана студентом за всі завдання, а також повна статистика групи загалом. Викладач бачить, який середній бал у групі з кожного завдання, скільки студентів намагалося вирішувати кожне завдання та скільки з них виконали це завдання на повний бал.

Кількість максимальних балів можна встановлювати для кожного завдання окремо, і ця кількість може бути довільною. Отже, можна налаштувати систему оцінювання під будь-яку шкалу оцінок.

Значною перевагою цієї системи для викладача є змога переглянути вихідний код усіх рішень, які студенти відправили на перевірку. До того ж навіть якщо студент відправив повторно своє рішення за одним і тим самим завданням, у системі зберігаються всі його попередні рішення. У системі передбачено режим, в якому оцінка виставляється не автоматично, а лише після того, як викладач переглянув вихідний код програми та прийняв це рішення. Отже, можна контролювати не тільки відповідність результату еталонному, але й перевіряти, наскільки студент виконав додаткові вимоги до цього завдання, наприклад, реалізував саме той алгоритм або спосіб вирішення завдання, що визначив викладач. Крім того, якщо викладач дійшов висновку, що студент виконував завдання не самостійно або серйозно порушив вимоги до вирішення, то таке рішення викладач може скасувати.

Висновки

У статті обґрунтовано використання автоматизованих систем тестування для вивчення основ програмування та алгоритмізації. Проаналізовано наявні підходи та виявлено їхні переваги та недоліки. Автоматизованою системою тестування для організації перевірки рішень студентів у процесі вивчення дисциплін «Програмування» та «Алгоритми та структури даних» запропоновано використовувати систему DOTS, яка, на думку авторів, є найбільш ефективним інструментом для досягнення поставленої мети. Перевагами цієї системи є наявність великої бази завдань різного рівня складності, починаючи від найпростіших і до завдань олімпіадного характеру, широту охоплення тем за алгорит-

мами та структурами даних, легкість та зручність створення нових турнірів, наповнення їх завданнями та додавання до цих турнірів студентів, можливість автоматичної перевірки всіх завдань та формування підсумкових таблиць із результатами цієї перевірки; можливість для викладача контролювати всі етапи виконання студентами завдання, переглядати вихідний код програм, відправлених на перевірку та керувати всім процесом. Досвід застосування системи DOTS дозволяє з усією впевненістю стверджувати, що використання автоматизованої системи тестування у вивченні основ програмування й алгоритмізації у підсумку дозволяє ефективно формувати компетентності випускників у проєктуванні, розробленні та аналізі алгоритмів, а також реалізації алгоритмів у вигляді програмних та інформаційних систем.

Література

1. Computing Curricula 2020: Paradigms for Global Computing Education. URL: <https://dl.acm.org/doi/book/10.1145/3467967>.
2. Мелешко Є. В., Якименко М. С., Поліщук Л. І. Алгоритми та структури даних: навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. Кропивницький: Видавець – Лисенко В. Ф., 2019. 156 с.
3. Ткачук В. М. Алгоритми і структура даних: Навчальний посібник. Івано-Франківськ: Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.
4. Коротеєва Т. О. Алгоритми та структури даних: навч. посібник. Львів: Видавництво Львівської політехніки, 2014. 280 с.
5. Зубков О. В., Семичева Н. Л. Роль автоматизованих тестируючих систем в процесі формування навчальних алгоритмічного мислення // Педагогический ИМИДЖ. 2019. № 4 (45). С. 550–565.
6. Айдаров К. А., Балакаева Г.Т. Применение тестирующей онлайн системы для развития практических навыков обучающихся в программировании // Материалы 48-ой научно-методической конференции «Роль высших учебных заведений в модернизации общественного сознания: переход к модели “Университет 4.0”». 18–19 января 2018 г. Алматы: Каз. ун-т, 2018. С. 17–20.
7. Гаспарян А. В., Тимошина Н. В. Особенности автоматизации проверки задач по программированию // ИТ-портал. 2018. № 2 (18). URL: <http://itportal.ru/-science/tech/osobennosti-avtomatizatsii-proverki/> (дата обращения: 02.07.2019).
8. Горчаков Л. В., Стась А. Н., Карташов Д. В. Обучение программированию с использованием системы Ejudge // Вестник Томского государственного педагогического университета. 2017. Вып. 9 (186). С. 109–112.
9. eJudge – система автоматической проверки программ. URL: <https://ejudge.ru/>
10. Школа программиста. URL: <https://acmp.ru/>
11. Интернет-портал организационно-методического обеспечения дистанционных олимпиад с программирования для одарованной молодежи. URL: <https://www.e-olymp.com>.
12. Алексеев А. В., Беляев С. Н. Подготовка школьников к олимпиадам по информатике с использованием вебсайта: учебно-методическое пособие для учащихся 7–11 классов. Ханты-Мансийск: РИО ИРО, 2008. 284 с.
13. Жуковский С. С. «E-olimp» – система автоматической проверки задач та проведення олімпіад з інформатики в інтернеті // Комп'ютер у школі та сім'ї. № 1 (65). 2008. С. 48–50.
14. Жуковский С. С. Використання Інтернет-порталу організаційно-методичного забезпечення «E-OLIMP» для підготовки одарованих школярів до олімпіади з інформатики // Комп'ютер у школі та сім'ї. 2010. № 8. С. 47–48. URL: http://nbuv.gov.ua/UJRN/komp_2010_8_14.
15. DOTS – онлайн-платформа для обучения программированию и развития алгоритмического мышления. URL: <https://dots.org.ua/>

References

1. Computing Curricula 2020: Paradigms for Global Computing Education. URL: <https://dl.acm.org/doi/book/10.1145/3467967>.
2. Meleshko Ye. V., Yakymenko M. S., Polishchuk L. I. Alhorytmy ta struktury danykh: Navchalnyi posibnyk dlia studentiv tekhnichnykh spetsialnostei dennoi ta zaochnoi formy navchannia. Kropyvnytskyi: Vydavets – Lysenko V. F., 2019. 156 s.
3. Tkachuk V. M. Alhorytmy i struktura danykh: Navchalnyi posibnyk. Ivano-Frankivsk: Vydavnytstvo Prykarpatskoho natsionalnoho universytetu imeni Vasyliia Stefanyka, 2016. 286 s.
4. Korotieieva T. O. Alhorytmy ta struktury danykh: navch. posibnyk. Lviv: Vydavnytstvo Lvivskoi politekhniki, 2014. 280 s.
5. Zubkov O. V., Semicheva N. L. Rol' avtomatizirovanyh testiruyushchih sistem v processe formirovaniya navykov algoritmicheskogo myshleniya // Pedagogicheskij IMIDZH. 2019. № 4 (45). S. 550–565.
6. Ajdarov K. A. Balakaeva G. T. Primenenie testiruyushchej onlajn sistemy dlya razvitiya prakticheskikh navykov obuchayushchihsya v programmirovani // Materialy 48-oi nauchno-metodicheskoi konferencii «Rol' vysshih uchebnyh zavedenij v modernizacii obshchestvennogo soznaniya: perekhod k modeli «Universitet 4.0»». 18–19 yanvarya 2018 g. Almaty: Kaz. un-t, 2018. S. 17–20.

7. Gasparyan A. V., Timoshina N. V. Osobennosti avtomatizatsii proverki zadach po programmirovaniyu // ITportal. 2018. № 2 (18). URL: <http://itportal.ru/science/tech/osobennosti-avtomatizatsii-proverki/>
8. Gorchakov L. V., Stas' A. N., Kartashov D. V. Obuchenie programmirovaniyu s ispol'zovaniem sistemy Ejudge // Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta. 2017. Vyp. 9 (186). S. 109–112.
9. eJudge – sistema avtomaticheskoy proverki programm. URL: <https://ejudge.ru/>
10. SHkola programmista. URL: <https://acmp.ru/>
11. Internet-portal orhanizatsiino-metodychnoho zabezpechennia dystantsiinykh olimpiad z prohramuvannia dlia obdarovanoi molodi. URL: <https://www.e-olymp.com>.
12. Alekseev A. V., Belyaev S. N. Podgotovka shkol'nikov k olimpiadam po informatike s ispol'zovaniem veb-sajta: uchebno-metodicheskoe posobie dlya uchashchihsya 7–11 klassov. Hanty-Mansijsk: RIO IRO, 2008. 284 s.
13. Zhukovskiy S. S. “E-olimp” – sistema avtomaticheskoy perevirky zadach ta provedennia olimpiad z informatiky v interneti // Kompiuter u shkoli ta simi. №1 (65). 2008. S. 48–50.
14. Zhukovskiy S. S. Vykorystannia Internet-portalu orhanizatsiino-metodychnoho zabezpechennia “E-OLIMP” dlia pidhotovky obdarovanykh shkolariv do olimpiady z informatiky // Kompiuter u shkoli ta simi. 2010. № 8. S. 47–48.
15. DOTS – onlajn-platforma dlya obucheniya programmirovaniyu. URL: <https://dots.org.ua/>

Щербаків Олександр Всеволодович, к.т.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38050-636-20-09, oleksandr.shcherbakov@hneu.net,
Скорін Юрій Іванович, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38066-748-47-51, skorin.yuriy@gmail.com.

Improving the efficiency of learning programming and algorithmization through the use of an automated testing system

Abstract. Problem. One of the main fundamental disciplines in training of future IT-specialists are the disciplines “Programming” and “Algorithms and data structures”. The issue of testing the programs

developed by students becomes relevant. Obviously, the so-called manual testing is too time-consuming and inefficient, so the increasing number of teachers use automated testing systems. **Goal.** The goal is to analyze the existing approaches and find an effective solution to the problem of automated testing of student programs when studying the basics of programming and algorithmization. **Methodology.** The analysis carried out showed that there are two main approaches to the organization of automated testing of student programs. The first is the use of special information systems that automatically check the programs. The second is the use of Internet resources designed to use ready-made tasks in the process of learning or training. **Results.** The use of automated testing systems in the study of programming, algorithms and data structures was substantiated. The existing approaches were analyzed and their advantages and disadvantages were identified. It was proposed to use the DOTS system, which, according to the authors, is the most effective tool to achieve this goal. **Originality.** The advantages of the proposed approach are the presence of a large database of tasks of various levels of complexity, the ease and convenience of creating new tournaments, filling them with tasks and adding students to these tournaments, the ability to automatically check all tasks and generate final tables with the results. This gives the possibility for the teacher to control all stages of the assignment by students, view the source code of the programs sent for verification and manage the entire process. **Practical value.** The use of an automated testing system in the study of the basics of programming and algorithmization ultimately makes it possible to effectively form the competence of graduates in the design, development and analysis of algorithms, as well as in the implementation of algorithms in the form of software and information systems.

Key words: programming, algorithmization, automated testing system, efficiency of educational process.

Shcherbakov Oleksandr, Ph.D., Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets, tel. +38050-636-20-09, oleksandr.shcherbakov@hneu.net,
Skorin Yuriy, Ph.D., Assoc. Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets, tel. +38066-748-47-51, skorin.yuriy@gmail.com.