

УДК 004.4

*Царенко Д.Ю., Денисюк О.Р.***ІНФОРМАЦІЙНА ПІДСИСТЕМА ДЛЯ ОЦІНЮВАННЯ ЗНАТЬ СТУДЕНТІВ****ДВНЗ «Український державний хіміко-технологічний університет», м. Дніпро, Україна**

У роботі розглядаються процеси виконання тестування знань студентів, сучасні розробки, які оптимізують здійснення цих процесів, та запропоновано метод, що може покращити тестування. Питання якісної перевірки отриманих в процесі навчання знань є дуже актуальним в межах сучасної сфери освіти. Важливим є питання підготовки якісних завдань для тестування, тому що це потребує витрат великої кількості часу та зусиль викладача. Тому, як метод, що може спростити цей процес, в роботі розглядається генерація умов задач з декількох розділів теорії ймовірностей. Завдяки генерації, можна отримати дуже велику кількість варіантів умов для задач, що мають один шаблон, але різні числові значення. Процес генерації складається з випадкового вибору одного з декількох питань до умови задачі, окремої генерації числових значень і поєднання отриманих значень в єдину умову. Ймовірність отримати збіг кожного значення при генерації є дуже малою, і це зменшує ризик вирішення задачі за допомогою сторонньої допомоги. Також для зменшення ризику запропоновано створити у програмному забезпеченні таймер для обмеження часу на відповідь. У роботі спроектовано та створено програмне забезпечення, що допомагає оптимізувати процес перевірки знань та зменшити навантаження на викладача при створенні завдань для тестування. В процесі проектування були виявлені усі необхідні вимоги до створюваного програмного забезпечення. Також у роботі розглянуто існуючі інформаційні системи, що надають викладачам інструмент для створення завдань на різні тематики, їх переваги та недоліки. Наведені результати дослідження допомагають визначити актуальність використання програми з функцією генерації умови.

Ключові слова: тестування, генерація, теорія ймовірностей, програмне забезпечення, оптимізація, проектування, інформаційна система, автоматизація.

DOI: 10.32434/2521-6406-2019-6-2-91-96

Постановка проблеми

На сьогоднішній день контроль якості вищої освіти є дуже актуальною проблемою для всієї України. Зокрема це відноситься до спеціальностей, що пов'язані з інформаційними технологіями та інформаційними системами. Це обумовлено швидким розвитком цих галузей в сучасному світі та широким використанням їх в усіх сферах людської діяльності. Процеси перевірки отриманих студентами знань є дуже важливим аспектом підготовки компетентних спеціалістів. Актуальною проблемою для викладачів є створення таких завдань, які зможуть точно оцінити рівень підготовки та передбачити можливість використання студентами сторонніх джерел інформації. Складання достатньої

кількості унікальних завдань для перевірки знань є досить трудомістким процесом. Саме тому з розвитком інформаційних технологій та автоматизації багатьох сфер людської діяльності постало питання про створення програмного забезпечення, яке дозволило би спростити процес підготовки завдань для тестування та забезпечити унікальні умови задач.

Одним з методів, який може допомогти досягнути вищевказаної мети, є автоматична генерація умови завдань. Завдяки використанню даного методу, можна створити цільну умову задачі з включенням в формулювання всіх необхідних числових значень.

Це дозволить мінімізувати участь людини в процесі підготовки завдань для тестування, та,

завдяки генерації значень в заданих діапазонах, суттєво зменшити ймовірність отримання двох однакових умов. Тому автоматична генерація завдань є одним із найбільш доцільних способів вирішення описаного ряду проблем.

Аналіз останніх досліджень і публікацій

На сьогоднішній день існує програмне забезпечення, що дозволяє здійснювати підготовку завдань для тестування та перевірки знань студентів. Наприклад, комплекс програмного забезпечення MyTestX має функціонал для створення завдань з різних дисциплін, з можливістю використання різних типів завдань. Його можливості включають в себе також виконання одночасного тестування групи студентів по локальній мережі [1]. Більш складний комплекс TestMaker, окрім зазначених функцій, дозволяє створювати тестові завдання, градуюючи їх за рівнями важкості [2]. Наведені комплекси програмного забезпечення, проте, потребують створення завдань викладачем, не дозволяючи автоматизувати цей процес.

Формулювання мети дослідження

Метою дослідження даної роботи є проектування та реалізація програмного забезпечення, яке, завдяки генерації умови задачі, допоможе автоматизувати процеси підготовки завдань для тестування студентів та дозволить підви-

щити якість перевірки знань, завдяки унікальності отриманих задач. Програма має включати в себе простий та інтуїтивно зрозумілий інтерфейс користувача. Крім створення завдань, розроблений програмний продукт повинен перевіряти відповіді студентів і виводити результати оцінювання.

Виклад основного матеріалу дослідження

Для демонстрації можливостей підходу до розробки програмного забезпечення для тестування студентів, заснованого на автоматичній генерації умови задачі, було використано стандартні задачі з дисципліни «Теорії ймовірностей». Курс теорії ймовірностей викладається на більшості технічних спеціальностей, тому розроблена програма має досить широку область застосування.

Перед початком розробки програмного продукту, що відповідає усім зазначеним вище вимогам, потрібно створити проект системи [3,4]. Для цього можна використовувати ErWin Process Modeler, що дозволяє представити систему, яка розробляється, у вигляді набору взаємопов'язаних функцій та декомпонувати кожен з них до необхідної глибини [5]. Згідно з зазначеними вимогами, програма повинна генерувати умову задачі разом із чисельними значеннями, приймати відповіді від студента, роз-

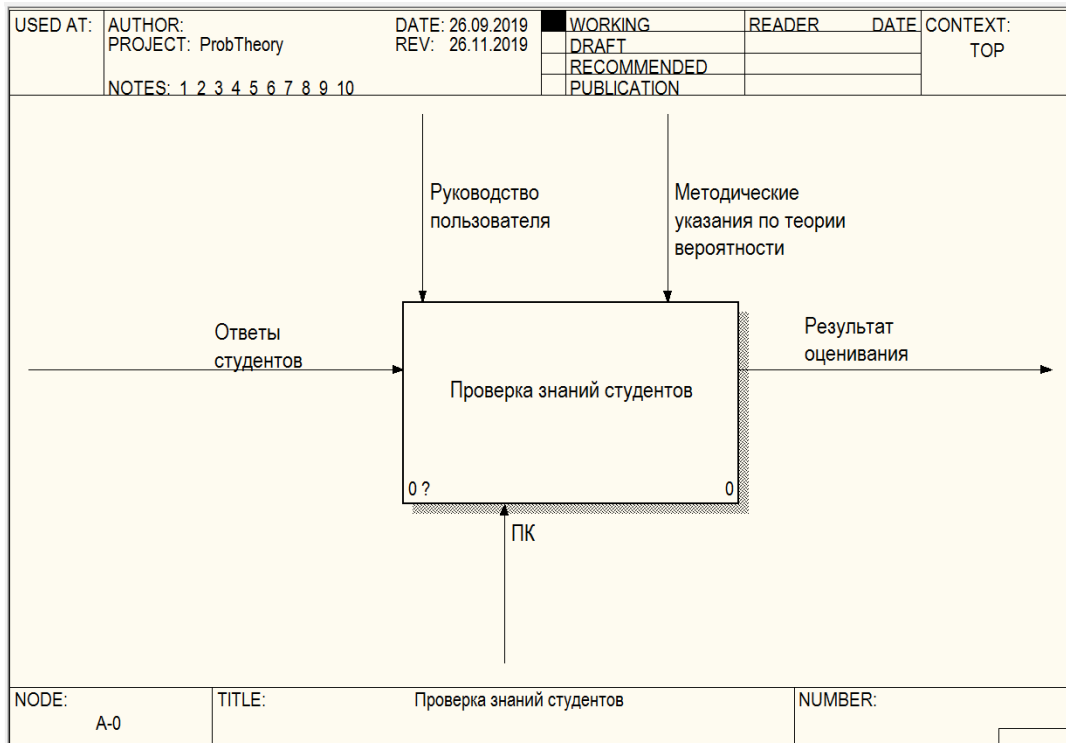


Рис. 1. Контекстна діаграма A0

в'язувати згенеровану задачу та порівнювати отриманий результат з відповіддю студента. Після закінчення тестування необхідно вивести на екран кількість правильних і неправильних відповідей. Для обмеження часу на розв'язок задачі, потрібно передбачити в програмі таймер. Якщо час на таймері вийшов раніше, ніж студент надав відповідь, необхідно перейти до наступної задачі та зарахувати попередню як розв'язану неправильно. З урахуванням всіх вищевказаних умов, було створено контекстну діаграму системи (рис. 1).

Відповідно із контекстною діаграмою на вхід програми поступають відповіді студентів. Вихідною інформацією є результат тестування. В свою чергу, засобами управління є керівництво користувача та методичні вказівки з теорії ймовірностей. Виконавчий механізм в даній контекстній діаграмі – персональний комп'ютер. Для детальнішого зображення роботи програми та коректного опису виконуваних функцій необхідно провести декомпозицію контекстної діаграми (рис. 2).

В результаті декомпозиції можна побачити, що процес перевірки знань студентів складається з чотирьох основних робіт: генерація умови, розв'язок задачі, перевірка відповіді та обробка результатів. Ці процеси пов'язані між

собою: з виходу першої роботи на вхід другої надходять згенеровані для умови значення, з другої на третю – правильна відповідь до цієї задачі, з третьої на четверту – мітка відповіді – деякий маркер, що позначає, правильну чи неправильну відповідь вказав студент при розв'язанні задачі. Після обробки усіх результатів на виході відповідної роботи отримується результат тестування. Далі потрібно виконати декомпозицію отриманих процесів для їх більш детального пояснення. На рис. 3 надано результат декомпозиції процесу генерації умови, на рис. 4 – результат декомпозиції процесу обробки результатів. Декомпозицію робіт виконано з використанням нотації IDEF3 для того, щоб показати послідовність виконання цих процесів.

Декомпозиція процесу генерації умови надана у вигляді послідовності дій. Спочатку генерується випадкове число, яке є номером одного з декількох питань, потім, у залежності від обраного числа, обирається питання, далі генеруються усі числові значення та вони поєднуються в єдину умову. Після цього числові значення використовуються при остаточному формулюванні умови завдання.

Декомпозиція процесу обробки результатів надана в вигляді послідовності дій. Після того, як студент відповідь послідовно на всі питан-

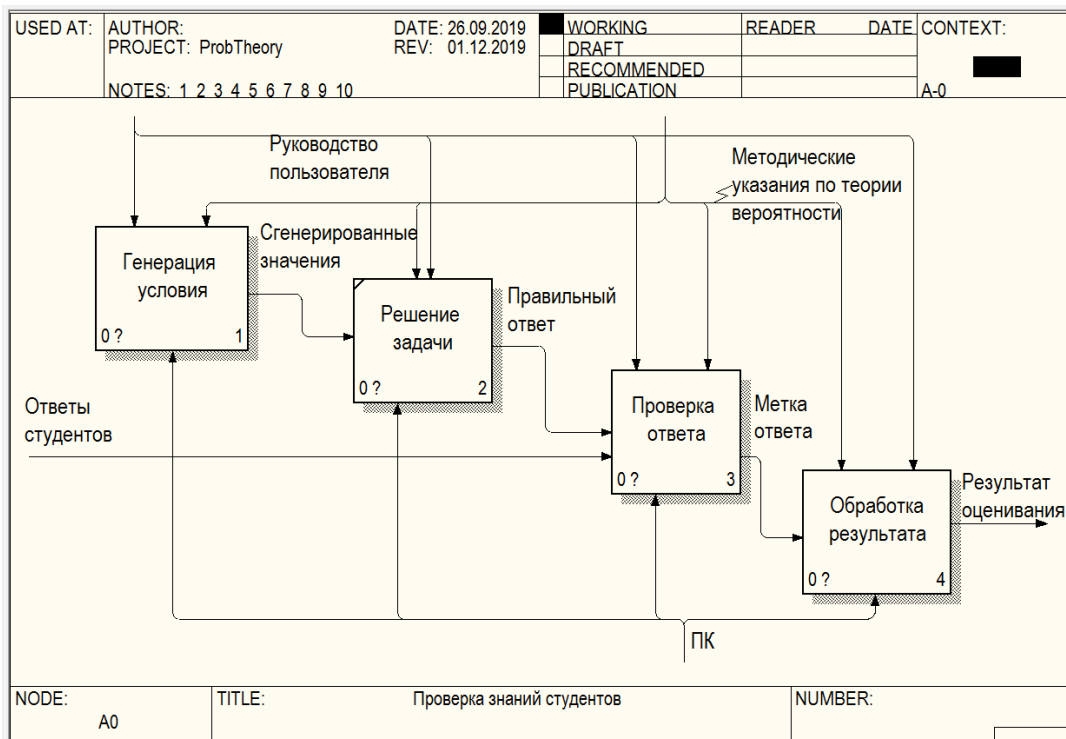


Рис. 2. Результат декомпозиції контекстної діаграми

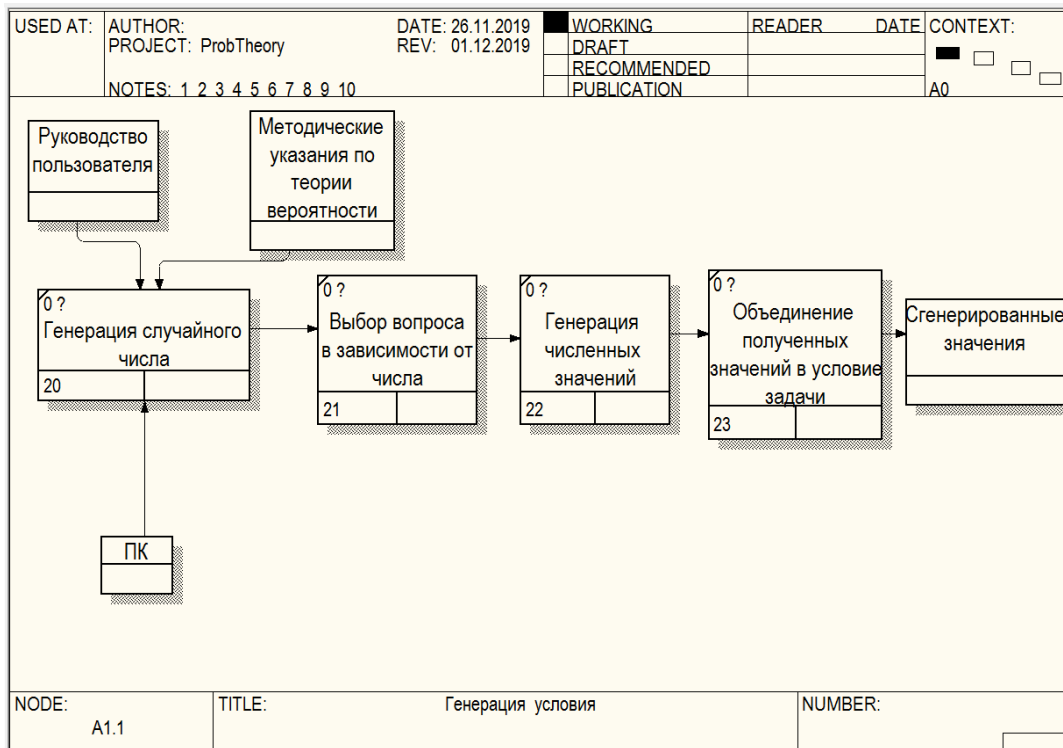


Рис. 3. Результат декомпозиції процесу генерації умови

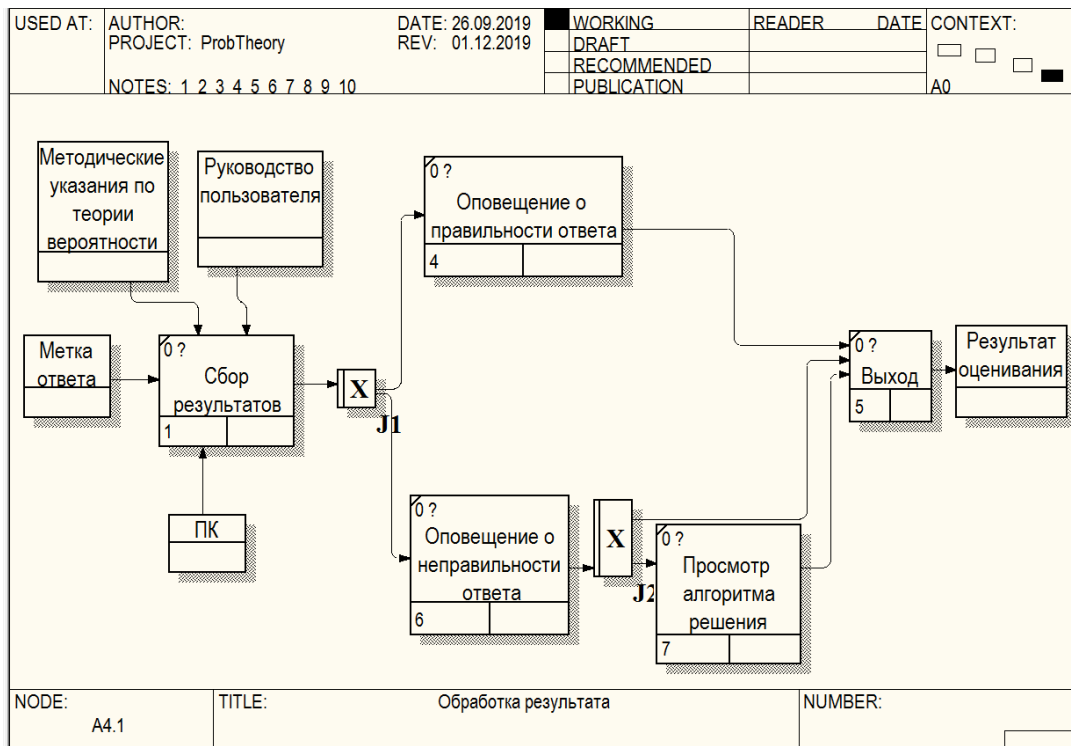


Рис. 4. Результат декомпозиції процесу обробки результатів

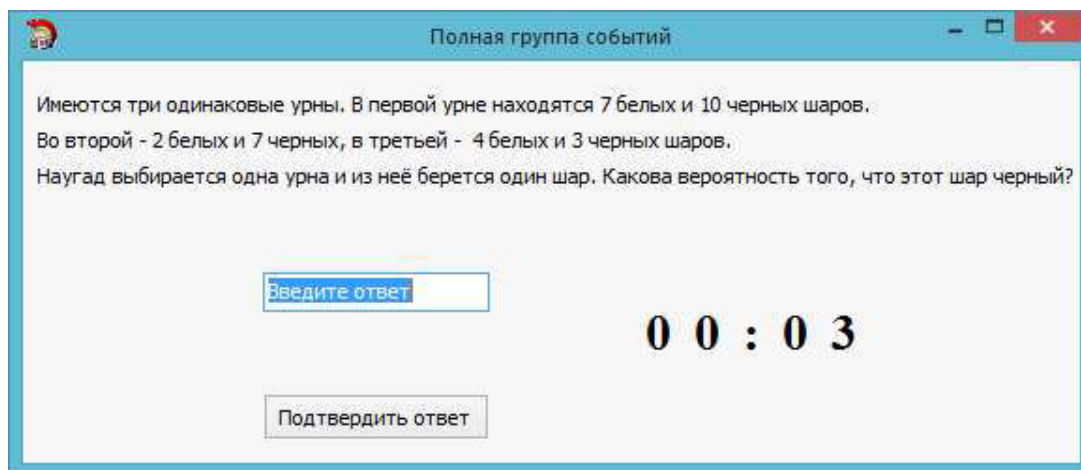


Рис. 5. Результат работы программы

ня, програма надає можливість переглянути кількість правильних та неправильних відповідей. У випадку, якщо відповідь буде неправильною, у студента буде можливість переглянути алгоритм розв'язання задач за даною темою для перевірки своїх знань.

Таким чином, інформації, що отримана під час проектування, достатньо для створення програмного продукту. Окрім зазначених вище вимог, програма повинна мати зрозумілий інтерфейс користувача. Для виконання даного завдання обрано мову програмування C++ [6]. Інтерфейс користувача розробленої програми можна побачити на рис. 5.

В даному випадку надано автоматично згенеровану задачу на тему «Повна група подій» [7]. В процесі генерації, в умові обираються чисельні значення та питання – чорний шар було обрано з урни чи білий. В даній задачі є сім значень, що генеруються. Чисельні значення обираються із діапазону від 2 до 9. В результаті, ймовірність отримати дві однакові умови дорівнює 0,0000005. Також на цій формі присутні поле для введення відповіді, кнопка, при натисканні якої відповідь буде підтверджена та передана до програми, та таймер. Якщо студент не встигає дати відповідь за відведений час, поточна форма закриється і відкриється наступна, а відповідь буде зараховано як неправильну. Завдяки отриманим результатам, можна стверджувати, що використання подібного програмного забезпечення може автоматизувати процес перевірки знань студентів.

Висновки

В даній роботі розглянуто актуальність питання якісної перевірки знань студентів вищих навчальних закладів, та проблематику підготов-

ки завдань для тестування. Запропоновано використання програмного забезпечення з можливістю генерування умови задачі та оцінювання знань студентів. Переглянуто переваги та недоліки існуючих систем. Зроблено проектування інформаційної системи, розглянуто усі аспекти розробленого програмного забезпечення. Відповідно до вимог, створено необхідне програмне забезпечення. Наведено результати виконання роботи.

СПИСОК ЛІТЕРАТУРИ

1. *Сипайло С.В.* Текущий контроль знаний студентов с помощью системы электронного тестирования MyTestX // Труды БГТУ. – 2014. – № 8. – С.145-146.
2. *Ильина Т.В., Кузьмин А.И., Мамонова В.С.* Обзор возможностей программных комплексов TestMaker и VeralTestEditor // Образовательные технологии и общество. – 2017. – № 2. – С.346-354.
3. *Ляшенко О.А., Шулак В.О.* Застосування функціонально-модульного підходу до проектування підсистеми визначення збалансованого раціону харчування // Комп'ютерне моделювання: аналіз, управління, оптимізація. – 2018. – № 2 (4). – С.34-44. <https://doi.org/10.32434/2521-6406-2018-4-2-34-44>
4. *Солодка Н.О., Ляшенко О.А.* Проектування та розробка клієнт-серверного додатку на основі однорангових мереж / Н.О. Солодка, О.А. Ляшенко // Математичне моделювання: Науковий журнал. – Кам'янське: ДДТУ, 2016. – № 2(35). – С. 87-93.
5. *Flowers R., Edeki Ch.* Business Process Modeling Notation // International Journal of Computer Science and Mobile Computing. – 2013. – Vol.2. – P.35-40.
6. *Borland C++ Builder 6.* Руководство разработчика / Д. Холлингворт, Б. Сворт, М. Кэшман, П. Густавсон. – Пер.

с англ. — М.: Издательский дом «Вильямс», 2003. — 976 с.

7. Крупин В.Г., Павлов А.Л., Попов Л.Г. Высшая математика. Теория вероятностей, математическая статистика, случайные процессы. Сборник задач с решениями: учебное пособие. — М.: Издательский дом МЭИ, 2013. — 368 с.

Надійшла до редакції 19.11.2019

ИНФОРМАЦИОННАЯ ПОДСИСТЕМА ДЛЯ ОЦЕНКИ ЗНАНИЙ СТУДЕНТОВ

Царенко Д.Ю., Денисюк О.Р.

В работе рассматриваются процессы проведения тестирования знаний студентов, современные разработки, которые оптимизируют проведение этих процессов, и также предложен метод, который может улучшить тестирование. Вопрос качественной проверки, полученных в процессе обучения знаний, является очень актуальным в пределах современной сферы образования. Важным является вопрос подготовки качественных задач для тестирования, так как это требует затрат большого количества времени и усилий преподавателя. Поэтому, как метод, который может упростить этот процесс, в работе рассматривается генерация условий задач из нескольких разделов теории вероятностей. Благодаря генерации, можно получить очень большое количество вариантов условий для задачи, имеющие один шаблон, но различные числовые значения. Процесс генерации состоит из случайного выбора одного из нескольких вопросов к условию задачи, отдельной генерации числовых значений и сочетание полученных значений в единственное условие. Вероятность получить совпадение каждого значения при генерации очень мала, и это уменьшает риск решения задачи с помощью посторонней помощи. Также для уменьшения риска предложено создать в программном обеспечении таймер для ограничения времени ответа. В работе спроектировано и создано программное обеспечение, помогающее оптимизировать процесс проверки знаний и уменьшить нагрузку на преподавателя при создании задач для тестирования. В процессе проектирования были обнаружены все необходимые требования к создаваемому программному обеспечению. Также в работе рассмотрены существующие информационные системы, предоставляющие преподавателям инструмент для создания задач на различные тематики, их преимущества и недостатки. Приведенные результаты исследования помогают определить актуальность использования программы с функцией генерации условия.

Ключевые слова: тестирование, генерация, теория вероятностей, программное обеспечение, оптимизация, проектирование, информационная система, автоматизация.

INFORMATION SUBSYSTEM FOR ASSESSING KNOWLEDGE OF STUDENTS

Tsarenko D. Yu., Denysiuk O. R.

Ukrainian State University of Chemical Technology, Dnipro, Ukraine

This paper examines student knowledge testing processes, current developments that optimize these processes, and suggests a method that can improve testing. The question of qualitative verification of knowledge acquired in the process of learning is very relevant within the modern sphere of education. It is important to prepare qualitative tasks for testing because it requires a lot of time and effort on the part of the teacher. Therefore, as a method that can simplify this process, the paper considers the generation of task conditions from several sections of probability theory. Due to the

generation, you can get a lot of variants of the task conditions, which have one pattern but different numerical values. The generation process consists of randomly selecting one of several questions to a task condition, separately generating numeric values, and combining the obtained values into a single condition. The likelihood of each match being generated at generation is very small, and this reduces the risk of the problem being solved with the help of a third party. Also, to reduce the risk, it is suggested to create a timer in the software to limit the response time. The software is designed and created to help streamline the knowledge assessment process and reduce the burden on the teacher when creating test tasks. During the process of designing all the necessary requirements for the created software were identified. The paper also examines existing information systems that provide teachers with the tools to create assignments on a variety of topics, their advantages and disadvantages. The results of the study help to determine the relevance of the program with the function of generating a condition.

Keywords: testing, generation, probability theory, software, optimization, designing, informational system, automation.

REFERENCES

1. Sipajlo S.V. Tekushhij kontrol' znaniy studentov s pomoshh'ju sistemy jelektronnogo testirovaniya MyTestX [Current control of students' knowledge using the MyTestX electronic testing system] *Trudy BGTU*. 2014, no. 8, pp.145-146. (in Russian).
2. Il'ina T.V., Kuz'min A.I., Mamonova V.S. *Obzor vozmozhnostej programmnyh kompleksov TestMaker i VeralTestEditor* [Overview of the capabilities of TestMaker and VeralTestEditor software systems]. *Obrazovatel'nye tehnologii i obshchestvo* [The educational technologies and society], 2017, no. 2, pp.346-354. (in Russian).
3. Liashenko O.A., Shulak V.O. *Zastosuvannja funkcional'no-modul'nogo pidhodu do proektuvannja pidsistemi viznachennja zbalansovanogo racionu harchuvannja* [Application of a functional-modular approach to designing a subsystem for determining the balanced diet]. *Komp'uterne modeljuvannâ: analiz, upravlinnâ, optimizaciâ* [Computer Modeling: Analysis, Control, Optimization], 2018, № 2 (4), pp.34-44. (in Ukrainian). <https://doi.org/10.32434/2521-6406-2018-4-2-34-44>
4. Solodka N.O., Liashenko O.A. *Proektuvannja ta rozrobka klient-servernogo dodatku na osnovi odnorangovih merezh* [Design and development of a client-server application based on peer-to-peer networks] *Matematichne modeljuvannja: Naukovij zhurnal* [Mathematical Modeling: A Scientific Journal], Kam'jans'ke, DDTU, 2016, № 2(35), pp.87-93. (in Ukrainian).
5. Flowers R., Edeki Ch. *Business Process Modeling Notation*. International Journal of Computer Science and Mobile Computing, 2013, vol.2, pp.35-40.
6. Hollingvort D., Svort B., Kjeshmjén M., Gustavson P. *Borland C++ Builder 6. Rukovodstvo razrabotchika* [Borland C++ Builder 6. Developer's Guide]. Per. s angl. Moscow, Williams Publishing House, 2003, 976 p. (in Russian).
7. Krupin V.G., Pavlov A.L., Popov L.G. *Vysshaja matematika. Teorija verojatnostej, matematicheskaja statistika, sluchajnye processy. Sbornik zadach s reshenijami: uchebnoe posobie*. [Higher mathematics. Probability theory, mathematical statistics, random processes. Collection of problems with solutions: a tutorial]. Moscow, Publishing House MPEI, 2013, 368 p. (in Russian).