

УДК 004.42

*М. А. ПАЗУШКО, К. В. ІВАНЧЕНКО, Д. А. ІВАНЧЕНКО***МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО ДОДАТКУ  
АВТОМАТИЗАЦІЇ ПРОЦЕСУ ТЕСТУВАННЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SPRING**

DOI: 10.20998/2078-9130.2022.2.270129

Метою даної роботи було проєктування та створення інформаційної системи оцінки та тестувань знань студентів, яка б дозволила спростити процес формування тестів та оцінки знань студентів для викладачів. В роботі було сформовано представлення щодо поняття тестування знань. Розкрито принципи реалізації даного процесу. Було проведено огляд існуючих програмних рішень комп'ютерних навчальних програм, проаналізовано ринок існуючих аналогів та на основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до програмного продукту, що розробляється. Виходячи з поставлених вимог, було обрано технічні та програмні засоби для розробки програми, а саме мова програмування Java та фреймворк Spring, компоненти якого дозволяють збільшити продуктивність програмного продукту, а також спростити та пришвидшити налаштування, конфігурацію та запуск проєкту. Для досягнення поставленої мети в роботі вирішуються наступні задачі: обробка даних користувача, створення особистого кабінету, інформація про студентів та викладачів, створення та проходження тесту, оцінка якості знань студентів та створених викладачем тестів. Побудовано математичну, інформаційну та програмну модель системи. Була спроектована та побудована база даних. Розроблено структурну та функціональну схеми системи. Виділено три ролі користувачів: викладач, студент та незареєстрований користувач. Реалізоване збереження результатів тестування та можливість отримання математичної статистики результатів тестування. Велику роль грає забезпечення простоти та комфорту користування продуктом користувача. Тому при розробці програми багато уваги було приділено створенню дружнього та ергономічного інтерфейсу. Результатом було отримано інформаційну систему автоматизованого тестування, яка виконує поставлену задачу, а також відповідає усім вимогам і може бути впроваджена на практиці. Результати роботи можуть бути використані як для подальшого введення додатку у навчальний процес, так і для набуття студентами практичних навичок.

**Ключові слова:** моделювання, база даних, оцінювання знань, проєктування, java, spring, mysql.

*М. А. PAZUSHKO, K. V. IVANCHENKO, D. A. IVANCHENKO***MATHEMATICAL AND SOFTWARE DEVELOPMENT OF A TESTING AUTOMATION  
APPLICATION USING THE SPRING FRAMEWORK**

The purpose of this work was to design and create an information system for evaluating and testing students' knowledge, which would simplify the process of forming tests and evaluating students' knowledge for teachers. The paper formed an idea about the concept of knowledge testing. The principles of implementation of this process are revealed. An overview of existing software solutions for computer training programs was conducted, the market of existing analogues was analyzed and, based on the analysis, functional and non-functional requirements for the software product being developed were formed. Based on the requirements, technical and software tools were chosen for the development of the program, namely the Java programming language and the Spring framework, the components of which allow to increase the productivity of the software product, as well as to simplify and speed up the setup, configuration and launch of the project. To achieve the goal, the following tasks are solved in the work: processing user data, creating a personal account, information about students and teachers, creating and passing a test, evaluating the quality of students' knowledge and teacher-created tests. A mathematical, information and software model of the system was built. The database was designed and built. The structural and functional scheme of the system has been developed. There are three user roles: teacher, student and unregistered user. Saving test results and the ability to obtain mathematical statistics of test results have been implemented. An important role is played by ensuring the ease and comfort of using the product for the user. Therefore, when developing the program, a lot of attention was paid to creating a friendly and ergonomic interface. The result was an information system of automated testing that fulfills the task, and also meets all the requirements and can be implemented in practice. The results of the work can be used both for the further introduction of the application into the educational process, and for students to acquire practical skills.

**Keywords:** modeling, database, knowledge assessment, design, java, mysql.

**Вступ.** В умовах значного зросту попиту на програмне забезпечення в освітньому процесі під час пандемії COVID-19, виникла необхідність у розробці додатків, що допомагають організувати освітній процес онлайн. Серед систем управління електронним навчанням окремо можна виділити системи тестування знань. Для управління якістю освіти

необхідно мати надійний інструмент для її вимірювання та оцінки [1-3]. Особливо важливо наповнення та форми цих засобів, як то: тестування не як форма контролю, а як засіб навчання; грамотна постановка завдань та оцінка їх складності; використання інформаційних технологій та побудова тестів таким чином, щоб вони сприяли розвитку

© М. А. Пазушко, К. В. Іванченко, Д. А. Іванченко 2022

аналітичних здібностей учнів.

**Постановка проблеми.** Хоча й існує безліч платформ та додатків для автоматизації тестування, проте не має уніфікованої, яка б використовувалася усіма. В ході роботи було визначено причини такого положення справ шляхом аналізу предметної області, а саме існуючих аналогічних додатків та використовуваних програм для тестування.

Електронне тестування, як метод оцінки знань, практично виключає суб'єктивізм викладача, а також зменшує часові затрати як на створення і проведення, так і на перевірку результатів тестування, у порівнянні із письмовим або усним контролем. Використання програмного забезпечення для тестування надає можливість охоплення великої кількості людей при тестуванні без зайвих витрат викладача на організацію. Таким чином, даний тип оцінки є найбільш оперативним, сучасним та об'єктивним.

**Аналіз останніх досліджень і публікацій.** На сьогодні на ринку представлена безліч різноманітних програм та засобів, які виробляють перевірку знань учнів, по різноманітним предметам. У більшості випадків, кожен викладач обирає найзручніший для себе спосіб тестування за допомогою обраної ним платформи. Оскільки всі вони різні, мають різний функціонал та форму, студентам необхідно пристосовуватися до кожної з них, що також викликає проблеми. Саме тому, у деяких вищих навчальних закладах, як то ХАІ, ХНУ ім. Каразіна, та, можливо, інші, у межах кафедр, або навіть цілих факультетів, створюють власне програмне забезпечення для проведення тестування. Всі подібні програми для навчальних закладів можуть бути написані різними мовами програмування: Delphi, C++, C#, Java та інші.

Різнноманітним аспектам тестування, створення і застосування навчальних і тестувальних систем на базі сучасних інформаційних технологій, питанням розробки баз даних і баз знань програмних систем перевірки рівня знань присвячені численні праці українських та закордонних авторів: Аванесова В. С., Башмакова І. А., Клайна П., Гагаріна О. О., Титенка С. В., Пасічника В. В., Тонконового В. М., Brusilovsky P., Schwarz, Weber. Більшість з них здійснювали дослідження в сфері проведення тестувань, наповнення бази тестових завдань за допомогою засобів підтримки ручного створення тестових завдань, безпеки процесу тестування і відтворення результатів [4-6].

Однією з найпоширеніших систем для дистанційного навчання (тестування зокрема) як в Україні, так і в інших країнах є Moodle.

Порівняльний аналіз деяких програм для електронного тестування наведений у [7], найважливіші частини приведені у табл. 1. При порівнянні програмних продуктів з'ясувалося, що всі вони мають високу функціональність, надійність та валідність. За результатами дослідження програми INDIGO та Moodle, хоча й є складними у вивченні, проте мають великий функціонал та методичні вказівки по роботі і є кращими за інші запропоновані. Хоча оцінювання є суб'єктивним.

Таблиця 1 – Порівняльний аналіз програмних продуктів, призначених для створення завдань тестового контролю

Найменування критерію	TestMaker	RichTest	MyTestXPt <sup>o</sup>	INDIGO	Moodle
Функціональність	+	+	+	+	+
Завершеність	+	+/-	+	+/-	+
Відновлюваність	+	+/-	+/-	+/-	+
Стійкість до відмов	+	+/-	+	+/-	+
Зручність вивчення	+	+	+	-	+/-
Простота використання	+/-	+	+	+/-	+
Ефективність	+/-	+/-	+	+	+
Зручність супроводу	-	+/-	+	+	+
Портативність	-	+/-	-	+	+/-

Проведений аналіз існуючих систем та алгоритмів показує, що більшість пов'язані із конкретними предметами та не можуть бути застосовані для усіх. Деякі мають недружній інтерфейс та загалом незручні у використанні.

У повній мірі для організації та проведення тестування підходить система Moodle, і вона рекомендована до використання. Проте, метою роботи є розробка власного додатку, і на основі усіх плюсів розглянутих програм можна виокремити принципи продукту, що розробляється. До того ж, як сам процес розробки, так і подальший супровід програми може використовуватися у освітніх цілях в межах інженерно-фізичного інституту та інших факультетів, пов'язаних із програмуванням та проектуванням.

#### Основна частина.

На першому етапі було визначено три основні ролі: викладач, студент та незареєстрований користувач). Необхідно було виділити функціонал для кожної з них. На рис. 1 наведено діаграму використання для ролі «Викладач».

Дизайн додатку являє собою MVC (Model-View-Controller) і реалізується n-рівневою архітектурою [8]. Концептуально n-рівнева архітектура є лінійною, однак дизайн MVC є трикутним: представлення надсилає оновлення до контролера, контролер оновлює модель, а представлення оновлюється із моделі.

Користувач подає команди програмі. Контролер отримує ці команди та трансформує дані у моделі. Модель оновлюється і повідомляє про те, що потрібно змінити інтерфейс, щоб відобразити зміни в даних.

Патерн MVC допомагає розбити інтерфейс і серверний код на окремі компоненти. Таким чином набагато легше керувати та вносити зміни в обидві частини, незалежно одна від одної.

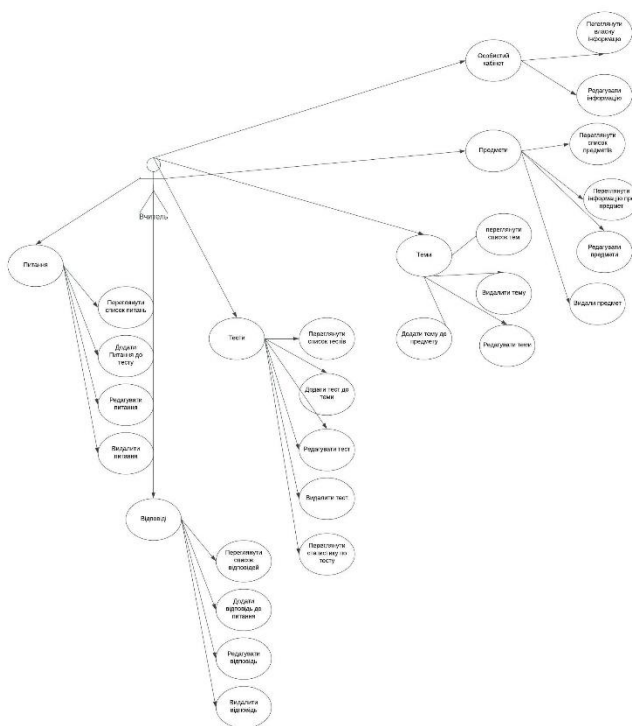


Рисунок 1 – Діаграма використання для користувача з роллю «Викладач»

Класична тривірнева система MVC складається з таких рівнів (рис. 2).

### Spring Boot MVC Project Structure

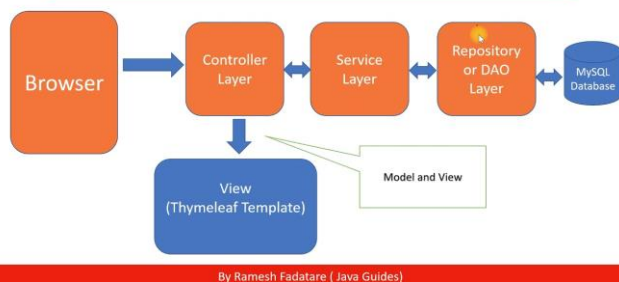


Рисунок 2 – Архітектура додатку, що розробляється на основі MVC

Рівень представлення (інтерфейс користувача), відповідно до назви, представляє дані користувачеві, візуалізує їх. Користувач взаємодіє тільки із цим рівнем. На даному рівні розташовані представлення і всі ті компоненти, які складають інтерфейс користувача (стилі, статичні сторінки html, javascript), а також моделі представлень, контролери, об'єкти контексту запиту. (приклад інтерфейсу користувача продемонстровано у додатку А).

Рівень бізнес-логіки містить набір контролерів, які регулюють обробку отриманих від рівня представлення даних, а також всю логіку програми, всі обчислення, взаємодіє з БД та передає рівню

представлення результат (приклад бізнес-логіки наведено у додатку Б). Є сполучною ланкою між представленням і моделлю, по суті являє собою «мозок» програми.

Рівень доступу до даних, він же модель. Моделі описують використовувані сутності, у них також розміщуються класи для роботи з різними технологіями доступу до даних. Незалежно від того, чи надходять дані з бази даних, API чи об'єкта JSON, модель відповідає за керування ними.

Особливу увагу було приділено проектуванню та реалізації бази даних. Процес проектування проходить у декілька етапів, а саме – аналіз предметної області, концептуальне проектування, логічне проектування та фізичне проектування. Після проведення аналізу предметної області, в ході якого було визначено зв'язки між об'єктами, а також визначені та описані атрибути, була розроблена концептуальна модель бази даних. Після цього було проведено перетворення концептуальної моделі у логічну, тобто приведення до кінцевого вигляду шляхом нормалізації. Логічна модель будується у термінах інформаційних одиниць, але без прив'язки до конкретної СУБД (рис. 3).

В контексті даної роботи математичне забезпечення програмного додатку являє собою статистичний аналіз отриманих результатів тестування. Основними характеристиками тесту загалом, які використовують системи є наступні: середня оцінка; медіана; коефіцієнт асиметрії; стандартна похибка; індекс легкості; коефіцієнт та ефективність дискримінації та інші.

Усі вони розраховуються за стандартними формулами статистики і надають інформацію про необхідність (або ні) корегування тестів для викладача. Даний «розділ» є не просто необхідним у подібних системах, але і обов'язковим, оскільки від якості складеного тесту залежить і якість освітнього процесу та його результати. Їх об'єктивна оцінка допоможе якомога точніше охарактеризувати засвоєння матеріалу учнями.

Використовувана модель математично-статистичного аналізу якості тестів і тестових завдань є традиційною в класичній теорії тестів. Вона являє собою розрахунок такого параметру як коефіцієнт надійності тесту за допомогою кореляції результатів. Коефіцієнт надійності ілюструє наскільки стійкими є показники тесту при повторному задіянні того ж тесту або його паралельного варіанту [9].

Варто зазначити, що даний коефіцієнт можна розраховувати декількома методами (ретестовий, паралельних форм та розщеплення). Оскільки найпоширенішим випадком є застосування тесту один раз на тій самій групі учнів, у програмі використовується саме метод розщеплення, тобто оцінити надійність лише при одному проходженні тесту.

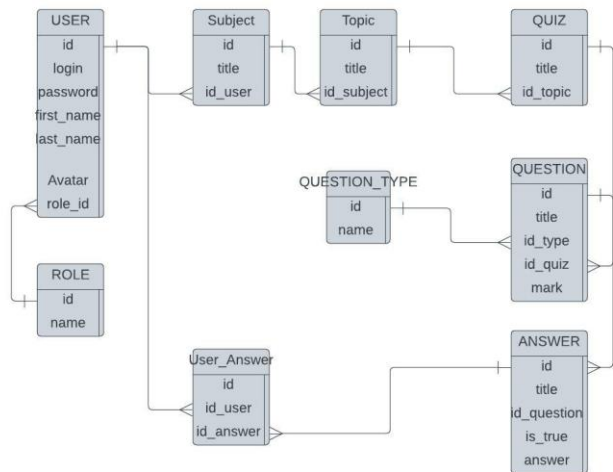


Рисунок 3 – Логічна модель бази даних

Метод розщеплення ґрунтується на припущенні, що тест є однорідним, тобто дві половини тесту є паралельними (хоча це не завжди є таким). Тому для більшої точності вводиться розрахунок валідності тесту. Тест може бути надійним, але не бути валідним, проте не може бути валідним, не будучи при цьому надійним. Надійність – необхідна умова валідності.

Валідність – характеристика здатності тесту вимірювати саме те, для чого він призначений, тобто відповідати своєму призначенню [10].

Етапи розрахунку надійності тесту:

1. Формування матриці результатів тестування.

Матриця ( $A$ ) формується наступним чином: строки ( $a_{nm}, n = \overline{1, N}$ ) – конкретні студенти, стовпці ( $a_{nm}, m = \overline{1, M}$ ) – питання тесту.

Відповідно, у клітинках містяться індикатори відповідей (результати відповідей 0 або 1). У результаті отримуємо:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \quad (1)$$

2. Реструктуризація матриці результатів.

Попередньо до розрахунків вводиться етап вибраковування. На даному кроці видаляються усі завдання із якими впоралися та не впоралися усі опитувані (усі стовпці де кожне значення дорівнює 0/1), оскільки ці завдання не вплинуть на результат. Також видаляємо кожного студента хто здав тест повністю, або не здав взагалі (усі строки, де кожне значення дорівнює 0/1).

Далі опитувані розташовуються у порядку спадання набраних балів (зверху вниз). Одночасно, стовпці переставляються за спаданням суми балів (зліва направо). В результаті вийде впорядкована матриця із умовною діагоналлю з лівого нижнього кута у правий верхній, з однієї сторони – одиниці, з іншої – нулі.

3. Розрахунок коефіцієнту надійності.

Наступним кроком ділимо отриману матрицю на

дві частини: перша ( $x$ ) – непарні стовпці, друга ( $y$ ) – парні. Для цього проводимо наступні розрахунки.

1 Середньоквадратичні відхилення для кожної із частин:

$$\sigma_x = \sqrt{D_x}, \quad (2)$$

$$\sigma_y = \sqrt{D_y}, \quad (3)$$

де  $D_x/D_y$  – дисперсія, що характеризує міру розсіяння балів студентів відносно середнього значення.

Дисперсія розраховується за формулами:

$$D_x = \overline{x^2} - \bar{x}^2 = \frac{\sum_{i=1}^n x_i^2}{n_1} - \frac{(\sum_{i=1}^n x_i)^2}{n_1^2}, \quad (4)$$

$$D_y = \overline{y^2} - \bar{y}^2 = \frac{\sum_{i=1}^n y_i^2}{n_1} - \frac{(\sum_{i=1}^n y_i)^2}{n_1^2} \quad (5)$$

де  $n_1$  – кількість завдань у кожній із частин.

Для означення міри розсіяння використовується саме значення середньоквадратичного відхилення, а не дисперсія оскільки остання виражається у квадратних одиницях;

2 Кореляційний момент:

$$k_{xy} = \overline{xy} - \bar{x} * \bar{y} = \frac{\sum_{i=1}^n (x_i y_i)}{n} - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n^2}. \quad (6)$$

3 Коефіцієнт кореляції:

$$r_{xy} = \frac{k_{xy}}{\sigma_x + \sigma_y}. \quad (7)$$

4 Коефіцієнт надійності:

$$r = \frac{2r_{xy}}{1+r_{xy}}. \quad (8)$$

Оскільки методом розщеплення довжина тесту змінюється, а саме скорочується вдвічі, то і отримане значення  $r_{xy}$  також буде зменшеним. Для його корегування і використовується формула (8). Чим вище значення коефіцієнтів кореляції, отриманих вказаними способами, тим вища надійність та валідність тесту [10].

Таким чином, по завершенню тесту викладач може отримати статистику, а саме число  $r \in (0,1)$ . Зазвичай, допустимим значенням називають 0.7, тобто якщо отримане значення менше – тест вважається не надійним і рекомендується змінити його. Але у нашому випадку рішення віддається на поручу викладачеві.

В процесі розрахунку надійності тесту використовуються дуже важливі параметри, які у розроблюваному додатку також подаються у графічному вигляді для розуміння викладача якості розробленого тесту, а саме: дисперсія (вона ж варіативність) та середньоквадратичне відхилення.

Варіація  $a_i b_i$ :

$$a_i = m/n, \quad (9)$$

що є часткою правильних відповідей на  $i$ -е завдання.

$$b_i = 1 - a_i, \quad (10)$$

відповідно, неправильних. У нормі значення варіативності складає приблизно 0.25.

Слабка варіація результатів свідчить про низьку якість тесту, так як свідчить про низьку диференціацію студентів за рівнем підготовки. Занадто високе значення дисперсії буває у випадку, коли всі опитувані відрізняються за кількістю виконаних завдань, і такий тест також вимагає редагування.

Таким чином, у розроблюваній програмі викладачеві надається статистика зображена на рисунку 4.

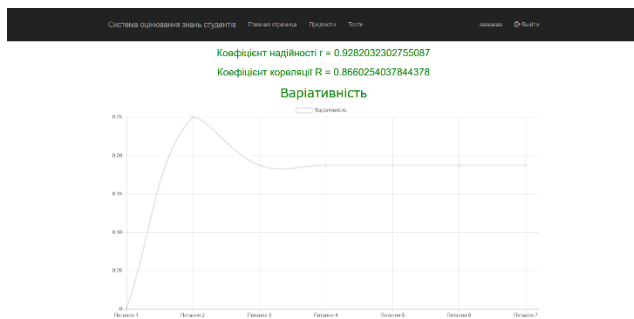


Рисунок 4 – Статистичний аналіз результатів тестування

У ході розробки додатку, як було зазначено вище використано фреймворк Spring Framework який використовує принципи MVC. Тому архітектура проекту виглядає наступним чином (рис. 4). Детальніше усі папки та їх вміст будуть розглянуті далі.

Докладніше, папка resources зберігає в собі файли зображень, налаштувань додатку (рис. 5), папка Controllers зберігає в собі контролери для доступу до даних з бази даних (рис. 6). Models – зберігає класи моделей (сутностей) додатку (рис. 7). Templates – зберігає у собі html сторінки інтерфейсу користувача. Config – папка з конфігураціями для поштового серверу та конфігурацію Spring Security. Repository, Service – папки з інтерфейсами та класами, що надають доступ до бази даних. Клас App – основна точка входу в програму.

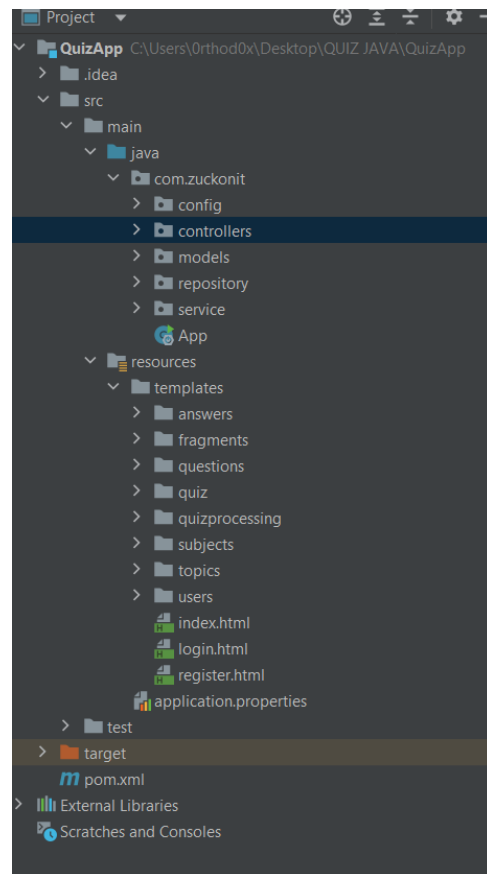


Рисунок 5 – Структура додатку

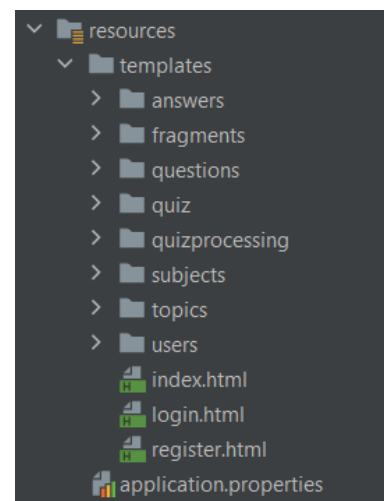


Рисунок 6 – Вміст папки resources

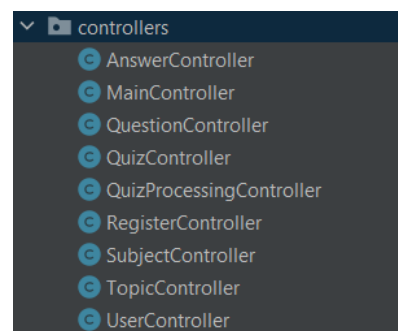


Рисунок 7 – Вміст папки Controllers

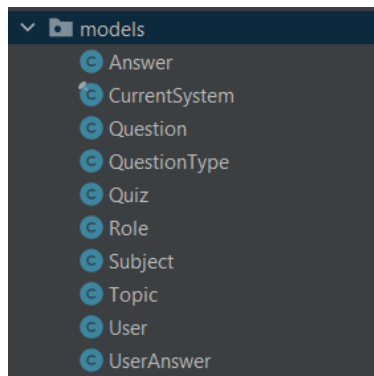


Рисунок 8 – Вміст папки Models

Окрім патернів, використовуваних Spring, для виконання специфічних завдань у додатку використовувались наступні патерни проектування: будівельник (builder) та синглтон (Singleton).

Будівельник — це породжувальний патерн проектування, що дає змогу створювати складні об'єкти крок за кроком. Будівельник дає можливість використовувати один і той самий код будівництва для отримання різних відображень об'єктів

Одинак — це породжувальний патерн проектування, який гарантує, що клас має лише один екземпляр, та надає глобальну точку доступу до нього [8].

Додаток був перевірений на сумісність із браузером (Chrome, Microsoft Edge, Firefox, Safari) та операційними системами (Linux, Windows)

Даний додаток тестувався мануальним способом в результаті якого були виявлені певні помилки при переходах на сторінки сайту, під час збереження та редагування даних, в результаті чого усі критичні помилки були виправлені, що дало змогу отримати правильні результати роботи додатку.

**Висновки.** Результатом було отримано інформаційну систему автоматизованого тестування, яка повноцінно виконує поставлену задачу, а також відповідає усім функціональним вимогам і може бути впроваджена на практиці. Результати роботи можуть бути використані як для подальшого введення додатку у навчальний процес, так і для набуття студентами практичних навичок.

#### Список літератури

1. Зайцева Л. В. *Моделі та методи адаптивного контролю знань* / Л. В. Зайцева, Н. О. Прокоф'єва // *Educational Technology & Society*. – 2004. – № 7 (4). – С. 265–277.

2. Катеринчук І. С. *Інтелектуальна система автоматизованого контролю знань студентів вищих навчальних закладів* / І. С. Катеринчук, Р. В. Рачок, В. В. Кравчук, В. М. Кулик // *Інформаційні технології в освіті: збірник наукових праць*. – Херсон : *Ізд-во ХГУ*, 2009. – Випуск 4. – С. 139–147.

3. S.H.P.W Gamage. *A systematic review on trends in using Moodle for teaching and learning* / S.H.P.W Gamage, J.R. Ayres, M.B. Behrend // *International Journal of STEM Education*. – 2022. – V.9.

4. Матвійчук А. О. *Інформаційна технологія гнучкого тестування рівня знань* / А. О. Матвійчук, О. В. Мазурець //

*Збірник наукових праць за матеріалами дев'ятої міжнародної науково-технічної конференції «Актуальні проблеми комп'ютерних технологій 2015»*. – Хмельницький, 2015. – С. 158–169.

5. Крак Ю. В. *Практичне дослідження ефективності інформаційної технології автоматизованого визначення семантичних термінів в контенті навчальних матеріалів* / Ю. В. Крак, О. В. Бармак, О. В. Мазурець // *Прикладне програмне забезпечення: збірник наукових праць за матеріалами десятої міжнародної науково-практичної конференції по програмуванню «УкрПРОГ'2016»*. – К., 2016. – С. 237–245.

6. Бармак О. В. *Інформаційна технологія автоматизованого визначення термінів у навчальних матеріалах* / О. В. Бармак, О. В. Мазурець // *Вимірвальна та обчислювальна техніка в технологічних процесах*. – Хмельницький, 2015. – № 2. – С. 94–102.

7. Андросов К.Ю. *Сравнительный анализ программ-конструкторов тестов и возможности их использования в учебном процессе* / К.Ю. Андросов // *«Ergodesign»* – 2019. – № 2. – С. 75–80.

8. Dinesh Rajput. *Spring 5 Design Patterns: Master efficient application development with patterns such as proxy, singleton, the template method, and more* / Rajput Diensh // *Packt Publishing Ltd.* – 2017. – С. 8.

9. *Актуальные вопросы языкового тестирования / под редакцией И.Ю.Павловской*. – СПб.: *Изд-во СПбГУ*, 2017. – Выход 2. – с.117.

10. Потеев М.И. *Статистическая обработка результатов тестирования и выработка рекомендаций по совершенствованию учебного процесса* / М.И.Потеев, Е.Г.Гой // *Образование и наука*. – 2002. – № 2. – с.101-108.

#### References (transliterated)

1. Zaitseva L. V. *Modeli ta metody adaptivnoho kontroliu znan [Models and methods of adaptive control of knowledge]* / L. V. Zaitseva, N. O. Prokofieva // *Educational Technology & Society*. – 2004. – № 7 (4). – s. 265–277.

2. Katerynychuk Y. S. *Intellektualnaya sistema avtomatizirovannogo kontrolya znaniy studentov vysshikh uchebnykh zavedeniy [Intelligent system for automated knowledge control of students of higher educational institutions]* / Y. S. Katerynychuk, R. V. Rachok, V. V. Kravchuk, V. M. Kulyk // *Informatsionnyie tehnologii v obrazovanii: sbornik nauchnykh trudov*. – Kherson: *Izd-vo HGU*, 2009. – Vyipusk 4. – S. 139–147.

3. S.H.P.W Gamage. *A systematic review on trends in using Moodle for teaching and learning* / S.H.P.W Gamage, J.R. Ayres, M.B. Behrend // *International Journal of STEM Education*. – 2022. – V.9.

4. Matviichuk A. O. *Informatsiina tekhnolohiia hnuchkoho testuvannia rivnia znan [Information technology for flexible knowledge level testing]* / A. O. Matviichuk, O. V. Mazurets // *Zbirnyk naukovykh prats za materialamy dev'ятої mizhnarodnoi naukovo-tekhnichnoi konferentsii «Aktualni problemy kompiuternykh tekhnolohii 2015»*. – Khmelnytskyi, 2015. – S. 158–169.

5. Krak Yu. V. *Praktychne doslidzhennia efektyvnosti informatsiinoi tekhnolohii avtomatyzovanoho vyznachennia semantychnykh terminiv v kontenti navchalnykh materialiv [A practical study of the effectiveness of information technology for the automated definition of semantic terms in the content of educational materials]* / Yu. V. Krak, O. V. Barmak, O. V. Mazurets // *Prykladne prohradne zabezpechennia: zbirnyk*

naukovykh prats za materialamy desiatoi mizhnarodnoi naukovykh konferentsii po prohramuvanni «UkrPROH2016». – K., 2016. – S. 237–245.

6. Barmak O. V. Informatsiina tekhnolohiia avtomatyzovanoho vyznachennia terminiv u navchalnykh materialakh [Information technology for automated definition of terms in educational materials] / O. V. Barmak, O. V. Mazurets // Vymiriuvalna ta obchysliuvalna tekhnika v tekhnolohichnykh protsesakh. – Khmelnytskyi, 2015. – № 2. – С. 94–102.

7. Androsov K.Iu. Sravnitelnyy analiz programm-konstruktorov testov i vozmozhnosti ih ispolzovaniya v uchebnom protsesse [Comparative analysis of test constructor programs and the possibility of their use in the educational process] / K.Iu. Androsov // «Ergodesign» – 2019. – № 2. – S. 75-80.

8. Dinesh Rajput. Spring 5 Design Patterns: Master efficient application development with patterns such as proxy, singleton, the template method, and more /Rajput Diensh// Packt Publishing Ltd. – 2017. – С. 8.

9. Aktualnyie voprosyi yazykovogo testirovaniya [Current issues of language testing]/pod redaktsiei Y.Iu.Pavlovskoi. – SPb.: Izd-vo SPbHU, 2017. – Vyipusk 2. – s.117.

10. Poteev M.Y. Statisticheskaya obrabotka rezultatov testirovaniya i vyirabotka rekomendatsiy po sovershenstvovaniyu uchebnogo protsessa [Statistical processing of test results and development of recommendations for improving the educational process] /M.Y.Poteev, E.H.Hoi // Obrazovanye y nauka. – 2002. – № 2. – s.101-108.

Надійшла (received) 24.12.2022

*Відомості про авторів / About the Authors*

**Пазушко Марина Андріївна** – Національний технічний університет «Харківський політехнічний інститут», магістр комп'ютерних наук; тел.: (099) 536-93-59; e-mail: Maryna.Pazushko@infiz.khpi.edu.ua.

**Pazushko Maryna Andriivna** – National Technical University "Kharkiv Polytechnic Institute", Master of Computer Sciences; phone: (099) 536-93-59; e-mail: Maryna.Pazushko@infiz.khpi.edu.ua.

**Іванченко Ксенія Вікторівна** – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри комп'ютерного моделювання процесів та систем; тел.: (097) 437-89-74; e-mail: [xeniya.ivanchenko@gmail.com](mailto:xeniya.ivanchenko@gmail.com), ORCID: <https://orcid.org/0000-0003-0960-1200>

**Ivanchenko Kseniia Viktorivna** – candidate of technical sciences, associate professor, National Technical University "Kharkiv Polytechnic Institute", associate professor of the Department of Computer Modeling of Processes and Systems; phone: (097) 437-89-74; e-mail: [xeniya.ivanchenko@gmail.com](mailto:xeniya.ivanchenko@gmail.com), ORCID: <https://orcid.org/0000-0003-0960-1200>

**Ivanchenko Dmytro Anatoliiovych** – candidate of technical sciences, Pryazovsky State Technical University, associate professor of the department of rolling stock of transport systems; phone: (096) 368-12-37; e-mail: [ukrsarтерps@gmail.com](mailto:ukrsarтерps@gmail.com). ORCID: <http://orcid.org/0000-0002-3024-3930>