

Москаленко А.М., Науменко Б.В., магістранти

Астістова Т.І., к.т.н., доц.

Київський національний університет технологій та дизайну

ДІДЖИТАЛІЗАЦІЯ ГУРТОЖИТКУ З ВИКОРИСТАННЯМ КЛІЄНТ-СЕРВЕРНИХ ТЕХНОЛОГІЙ JAVA ТА ФРЕЙМВОРКА ANGULAR

Анотація. Цифрові інновації – звична частина якісного навчання в усьому світі.

Для збереження конкурентоспроможності наших фахівців Україні вже сьогодні варто інвестувати в ІТ-рішення в систему освіти. Метою нашого проекту було створити соціально-освітню платформу на прикладі нашого університету. Наша платформа допоможе тисячам студентів правильно організувати свій час і власний навчальний простір у гуртожитку, а працівникам вищого навчального закладу уникнути кілометрових черг і проблематичних ситуацій. Вся проблематика, яка існує у гуртожитку не є безпідставною, вона тягнеться роками і саме створення нашого додатку може стати переломним моментом у відношенні до організації студентського життя. Наша платформа повністю перенесе деякі процеси, які існують у гуртожитку, у мобільний девайс чи персональний комп'ютер. Гуртожиток, як складова частина вищого навчального закладу, потребує інновацій. Архітектура нашої платформи для створення системи діджиталізації гуртожитку буде поділена на декілька модулів, які будуть працювати окремо і, в разі необхідності, зможуть обмінюватися даними. Можемо виділити наступні модулі: електронне поселення; автоматизація електроніки гуртожитку; модуль для оформлення запитів на комунальне обслуговування; модуль з внутрішньою соціальною мережею. Кожен модуль розробляється як окремий додаток, який вирішує свою задачу. Система технологічно буде мати серверну та клієнтську частини, які потребують взаємодії при впровадженні в проект.

Ключові слова: діджиталізація; REST-full; Java; Spring; MySQL; Angular; MVC; Node.js; REST.

Moskalenko A.M., Naumenko B.V., Astistova T.I.

Kyiv National University of Technologies and Design

DIGITALIZATION OF THE DORMITORY USING JAVA CLIENT-SERVER TECHNOLOGIES AND ANGULAR FRAMEWORK

Abstract. Digital innovation is a common part of quality education around the world. To maintain the competitiveness of our specialists, Ukraine should already invest in IT solutions in the education system. The aim of our project was to create a social and educational platform, on the example of our university. Our platform will help thousands of students to properly organize their time and their own study space in the dormitory, and employees of higher education to avoid long queues and problematic situations. All the problems that exist in the dormitory are not unfounded, it lasts for years and the creation of our application can be a turning point in relation to the organization of student life. Our platform will completely transfer some processes that exist in the hostel, to a mobile device or personal computer. The dormitory, as an integral part of higher education, needs innovation. We can distinguish the following modules: electronic settlement; automation of dormitory electronics; module for making utility requests; module with internal social network. Each module is developed as a separate application that solves its task. The system will technologically have server and client parts that require interaction during implementation in the project.

Keywords: digitalization; REST-full; Java; Spring; MySQL; Angular; MVC; Node.js; REST.

Вступ. Наш час – це епоха все більш широкого поширення цифрових технологій, які глибоко проникають в саму суть продуктів багатьох організацій і радикально змінюють природу інновацій в продуктах та послугах. Цифрові інновації – звична частина якісного навчання в усьому світі. Для збереження конкурентоспроможності наших фахівців, Україні вже сьогодні варто інвестувати в IT-рішення в систему освіти.

Сучасний навчальний процес у вищому навчальному закладі та сфери, які пов'язані з ним, все більше і більше потребує використання цифрових технологій.

Метою нашого проекту було створити соціально-освітню платформу, на прикладі нашого університету.

Наша платформа допоможе тисячам студентів правильно організувати свій час і власний навчальний простір у гуртожитку, а працівникам вищого навчального закладу уникнути кілометрових черг і проблематичних ситуацій. Вся проблематика, яка існує у гуртожитку не є безпідставною, вона тягнеться роками і саме створення нашого додатку може стати переломним моментом у відношенні до організації студентського життя.

Наша платформа повністю перенесе деякі процеси, які існують у гуртожитку, у мобільний девайс чи персональний комп'ютер.

Гуртожиток, як складова частина вищого навчального закладу, потребує інновацій. Ідея про діджиталізацію гуртожитка виникла саме через не оптимізовані внутрішні процеси, які проходять в гуртожитку.

Термін «діджиталізація» і перекладі означає «оцифрування», або «приведення в цифрову форму». Сьогодні вітчизняні науковці дедалі частіше використовують терміном «діджиталізація» тому, що воно охоплює більш широкий спектр значень. Наслідки переходу від аналогового до цифрового типу кодування інформації стали вивчатися не просто як технологічний, але й як соціальний, культурний та антропологічний процеси.

Постановка завдання. Отже, ми поставили перед собою завдання розробити систему яка відцифрує та автоматизує більшість елементів гуртожитку з метою економії часу студентів та працівників вищого навчального закладу в період поселення, а також економії електроенергії, фінансів, та покращення загального добробуту жителів гуртожитку.

Ми дослідили всі внутрішні процеси гуртожитку і виділили ряд проблем які можливо вирішити за рахунок діджиталізації цього процесу.

Архітектури нашої платформи для створення системи діджиталізації гуртожитку, буде поділена на декілька модулів, які будуть працювати окремо і в разі необхідності зможуть обмінюватися даними.

Можемо виділити наступні модулі:

- електронне поселення;
- автоматизація електроніки гуртожитку;
- модуль для оформлення запитів на комунальне обслуговування;
- модуль з внутрішньою соціальною мережею.

Кожен модуль розробляється як окремих додаток, який вирішує свою задачу. Для такої концепції дуже добре підходить таке новітнє рішення як REST-full архітектура. REST повністю змінив програмну інженерію після 2000 року.

Система технологічно буде мати серверну та клієнтську частини, які потребують взаємодії при впровадженні в проект.

Головною проблемою наряду з іншим, на наш погляд, є застарілий процес поселення та переселення в гуртожитках, який на даний момент не оптимізований і займає дуже багато часу.

Через відсутність автоматизації цього процесу, утворюються черги і великі скупчення людей, які не бажані (особливо в умовах карантину). Коменданти гуртожитків та деканати в цей період завантажені великим напливом студентів.

Наша платформа дозволить перенести цей процес у мобільний девайс чи персональний комп'ютер.

Кожен студент матиме змогу пройти всі етапи поселення:

- обрання кімнати;
- оформлення документів та подача їх до деканату в дистанційному режимі;
- онлайн черги на затвердження документів в деканаті та прийом у коменданта для вирішення особистих питань;
- швидка комунікація з головами старостату та студентами;

Також, актуальним питанням сьогодення постає автоматизація всієї електроніки в гуртожитку. Одним із пріоритетних моментів для кожного студента є доступ до всесвітньої мережі Інтернет, але нажаль отримати доступ до Інтернету в гуртожитку дещо проблематично.

Наша платформа зможе інтегрувати у собі весь процес підключення і оплати за послуги провайдера. Система сповіщень у нашому додатку буде нагадувати про щомісячну оплату, призупинення постачання послуг на період канікул та всі оновлення від постачальника Інтернету.

У гуртожитку є пральні машини, електронні плити. Але, нажаль існує постійні черги за використанням цих предметів. Це пов'язано з тим, що студенти невчасно забирають свої речі для прання та витрачають багато часу на пошук вільних пральних машин, бігаючи з поверху на поверх. Особливим болем для кожного студента є пошук готівки на оплату прання, бо зараз в пріоритеті оплата картою.

Електронні плити, які використовують студенти на кухнях, досить часто залишаються ввімкнутими без використання, а гірше, коли ті самі студенти забувають ще й про їжу на конфорці і це спричиняє дискомфорт всім жителям на поверсі.

Наш додаток вирішує ці проблеми за допомогою системи сповіщень, відстеження вільних місць на прання та готування їжі, інтеграція процесу оплати за прання. Хочемо підкреслити, така система буде регулювати трафік електроенергії та водопостачання, що принесе економічну користь.

Соціальна комунікація є невід'ємною ланкою при проживанні в гуртожитку. Це є не тільки перевагою в економії часу для користувачів нашої платформи, а й гарною підтримкою з боку старостату. Можна без проблем звернутися за допомогою на інтегрованому форумі платформи та отримати швидку відповідь; а рейтинг найкращих блоків дасть стимул кожній кімнаті відслідковувати свій статус по чистоті, та в разі необхідності подати скаргу на порушників порядку; сповіщення про головні івенти Ради Гуртожитку допоможуть студенту не прогавити важливу новину та тримати руку на пульсі в разі надзвичайних ситуацій.

Не можемо не додати такі переваги нашої платформи як:

- електронний запис до сантехніка та електрика по кліку однієї кнопки;
- в період підключення опалення діятиме форма для подання скарги про недостатнє опалення кімнати;
- сповіщення про заміну постільної білизни.

На сьогоднішній день існують додатки які частково реалізують функціонал даного стартапу і вже довели свою доцільність у використанні, але вони не використовують увесь свій потенціал на повну. Наприклад модульне середовище чудово справляється з організацією навчального процесу і довело свою ефективність під час дистанційного навчання. Дивлячись на успіх модульного середовища можна

зробити висновок про доцільність діджиталізації гуртожитку задля оптимізації ресурсів студента та працівників вищого навчального закладу.

Результати досліджень. Діджиталізація гуртожитка це складна, розподілена система, яка має серверну та клієнтську частини та потребує вагомому набору технологій та великої уваги до їх взаємодії та впровадження в проект. Підбір технологій варто починати із вибору мови програмування для серверної частини додатку, так як ця частина системи слугує «серцем» всієї платформи.

Завдяки дослідженням ринку додатків було прийнято рішення про необхідність створення кросплатформного додатку, бо через різноманітність операційних систем, серверів, що слугують для організації освітнього процесу неможливо обрати єдину систему, під яку буде розроблятися дане програмне рішення.

Розглянемо технології для розробки серверної частини системи.

Серед усіх кросплатформних мов програмування було обрано найпопулярнішу мову – Java. Однією з найбільших причин, чому Java настільки популярна, є незалежність платформи. Програми можуть працювати на декількох різних типах комп'ютерів; доки на комп'ютері встановлено середовище виконання Java (JRE), програма Java може працювати на ньому. Більшість типів комп'ютерів будуть сумісні з JRE, включаючи ПК, що працюють на Windows, Macintosh, Unix або Linux, а також великі мейнфрейми, а також мобільні телефони. Оскільки вона існує так довго, деякі найбільші організації у світі створені з використанням мови. Багато банків, роздрібних торговців, страхових компаній, комунальних підприємств та виробників використовують Java. Поки що не спостерігається ознак зниження використання Java. Натомість це мова, що розвивається, яка майже однозначно поєднує стабільність та інновації. Висвітлюючи довголіття мови, код, який ви, можливо, писали 15 років тому, працюватиме на найсучасніших JVM і отримає швидкісну перевагу останнього профілювання, перекладу власного коду та управління пам'яттю. Java принципово об'єктно-орієнтована. Код настільки надійний, оскільки об'єкти Java не містять посилань на зовнішні дані. Мова складається з 50 ключових слів, але Java API має тисячі класів з десятками тисяч методів, які ви можете використовувати у своїх програмах. Незважаючи на це, від розробників не очікується, що вони вивчать весь Java API, і більшість з них досвідчені та вільно володіють лише невеликою його частиною. Оскільки ми розроблюємо серверний додаток нам знадобиться технологія із стандартного набору Java – сервлет. Сервлет використовується для розширення можливостей серверів, що розміщують додатки, доступ до яких здійснюється за допомогою моделі програмування запит-відповідь. Хоча сервлети можуть відповідати на будь-який тип запиту, вони зазвичай використовуються для розширення програм, розміщених веб-серверами. Для таких програм технологія Java Servlet визначає специфічні для HTTP класи сервлетів.

Однак одних стандартних бібліотек не достатньо для того щоб реалізувати наш проект. Для створення легкого у підтримці та універсального коду нам потрібні певні надлаштування над мовою програмування, також відомі як фреймворки. Програмний фреймворк (англ. software framework) – це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Відповідно – програмний фреймворк може містити в собі також допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API. Для нашого проекту був обраний програмний каркас Spring. Spring – це потужний, легкий фреймворк, який використовується для розробки додатків. Якщо говорити ширше, можна сказати, що

фреймворк Spring - це чітко визначений інструмент, який підтримує кілька веб-додатків, що використовують Java як мову програмування. До запуску фреймворку в IT-секторі додатки розроблялись із використанням стандартів JEE. За допомогою цих стандартів ми можемо розгорнути додаток на будь-якому сервері додатків JEE.

Але у нього було кілька проблем, серед яких:

- код ускладнюється в процесі розробки додатку;
- продуктивність системи страждає через важкість під-програм;
- проблема пошуку компонента;

Ці проблеми були вирішені завдяки введенню фреймворку Spring. Spring став помітною на ринку завдяки основним характеристикам Spring, якими є її модульність. Тобто його можна розділити на різні модулі, кожен з яких обслуговує свою функціональність. Можна виділити такі основні переваги Spring:

1. Легкість. Spring Framework дуже легкий щодо своїх розмірів та функціональних можливостей. Аспектно-орієнтоване програмування (AOP).

2. Управління транзакціями. Це використовується для уніфікації декількох API управління транзакціями та використовується для координації транзакцій для об'єктів Java.

3. Контейнер. Структура Spring розробляє та управляє життєвим циклом та конфігураціями об'єктів програми.

4. Ін'єкція залежності – дозволяє розробляти вільно пов'язані додатки.

5. Інтеграція з іншими фреймворками. Spring не можна розглядати як окремих фреймворк, це скоріше сімейство фреймворків які об'єднуються у складну архітектуру. У нашому додатку нам знадобляться декілька таких фреймворків.

Для збереження інформації, додатку потрібна база даних, яка може містити в собі велику кількість інформації, зможе обслуговувати велику кількість користувачів одночасно та буде добре захищеною. Також вона повинна бути реляційною, щоб задовольнити потреби архітектури нашого додатку. Ми обрали СУБД MySQL.

Ми зробили огляд технологій, які використовуються у розробці серверної частини нашої системи. Проте сама серверна частина ніяк не допоможе користувачам.

Розробці клієнтської частини потрібно також приділити увагу. Важливо зробити вибір технологій для розробки користувацького інтерфейсу, який допоможе користувачу вільно орієнтуватися в додатку і приваблювати своїм дизайном нових користувачів. Для створення такого інтерфейсу потрібно потужні та сучасні рішення front-end, оскільки ми плануємо розробити веб-орієнтовану систему, до якої можна отримати доступ як із браузера будь-якого пристрою так і з мобільного додатку.

Для вирішення цієї задачі, звичайних рішень HTML, CSS та JavaScript недостатньо. Потрібні, як і у випадку із серверною частиною, певні налаштування та фреймворки. Проаналізувавши всі існуючі рішення було обрано фреймворк Angular.

Angular – це інтерфейс-фреймворк із відкритим кодом, розроблений Google для створення динамічних сучасних веб-програм. Вперше введена в 2009 році, ця структура за ці роки набрала величезних обертів для усунення непотрібного коду та забезпечення легших і швидших додатків.

Швидко еволюціонувавши від AngularJS у 2010 році до Angular 5 у 2017 році, сьогодні інтерфейсний фреймворк використовується понад 44,3% програмних інженерів для створення користувацьких інтерфейсів (Stack Overflow Developer Survey 2017). Angular допомагає створювати інтерактивні та динамічні односторінкові додатки (SPA) за допомогою своїх привабливих функцій, включаючи шаблонування, двосторонній зв'язок, модуляризацію, обробку RESTful API, введення залежностей та обробку AJAX. Дизайнери можуть використовувати HTML як мову шаблонів і навіть розширити синтаксис HTML, щоб легко передати компоненти програми. Вам не

потрібно покладатися на сторонні бібліотеки для створення динамічних додатків з Angular. Використовуючи цей фреймворк у своїх проектах, ви можете отримати багато переваг. Ми перелічили причини нижче:

- Підтримується Google.
- Використання мови TypeScript – а це налаштування для JavaScript, що забезпечує вищий рівень безпеки.
- Декларативний інтерфейс – використовує HTML для визначення інтерфейсу програми. HTML.
- POJO. Не потрібні додаткові функції отримання та встановлення. Кожен об'єкт, який Angular використовує, є POJO (Plain Old JavaScript Object),
- Легке тестування. У Angular тестування надзвичайно просто. Модулі Angular мають додаткові частини, якими легко керувати.
- Спрощений шаблон MVC. Кутова структура вбудована в оригінальну архітектурну програму MVC (Model-View-Controller). Angular не просить розробників розділити додаток на різні компоненти MVC та побудувати код, який може їх об'єднати (рис. 1).

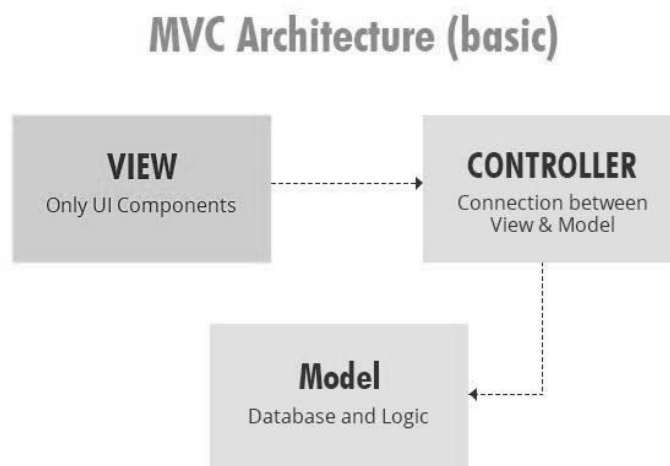


Рис. 1. Архітектура MVC

- Модульна структура. Angular організовує код у сегменти(модуль), будь то компоненти, директиви, канали чи послуги.
- Багаторазове використання. Структура Angular робить компоненти багаторазовими в додатку. Ви можете створювати користувацький інтерфейс з рухомими частинами, забезпечуючи при цьому плавний процес розробки.
- Спрощена модульна перевірка. Будучи незалежними один від одного, компоненти значно спрощують модульне тестування.
- Покращена читабельність. Послідовність у кодуванні робить читання коду задоволенням для нових розробників у поточному проекті, що підвищує їх продуктивність.
- Простота обслуговування. Незв'язані компоненти можна замінити кращими реалізаціями. Простіше кажучи, це забезпечує ефективне обслуговування та оновлення коду.

За допомогою Angular ми повністю розробимо клієнтську частину системи, яка за допомогою фреймворка, буде компілюватися як у веб так і у мобільний додаток. Працювати він буде у середовищі Node.js. Node.js – це міжплатформне середовище

виконання з відкритим кодом для розробки серверних та мережевих додатків. Програми Node.js написані на JavaScript і можуть запускатися в середовищі виконання Node.js в OS X, Microsoft Windows та Linux. Node.js також надає багату бібліотеку різних модулів JavaScript, що значною мірою спрощує розробку веб-додатків, що використовують Node.js. Node.js – це платформа, побудована на середовищі виконання Chrome, щоб легко створювати швидкі та масштабовані мережеві програми. Node.js використовує керовану подіями модель, що не блокує введення-виведення, що робить її легкою та ефективною, ідеально підходить для додатків у режимі реального часу, що працюють на розподілених пристроях.

Система діджиталізації гуртожитку буде поділена на декілька модулів.

Кожен модуль розробляється як окремий додаток який вирішує свою задачу. Для такої концепції дуже добре підходить таке новітнє рішення як REST-full архітектура. REST повністю змінив програмну інженерію після 2000 року. Цей новий підхід до розробки веб-проектів та послуг був визначений Роєм Філдіном, батьком специфікації HTTP і одним з провідних міжнародних авторитетів. У галузі API, REST (передача репрезентативного стану) сьогодні є передовою у розробці сервісних додатків. На сьогоднішній день не існує проектів чи програм, які не мають REST API для створення професійних служб на основі цього програмного забезпечення. Системи ідентифікації Twitter, YouTube, Facebook ... сотні компаній створюють бізнес завдяки REST та REST API. Без них будь-яке зростання було б практично неможливим. Це пов'язано з тим, що REST – це найбільш логічний, ефективний і поширений стандарт у створенні API для Інтернет-послуг. Щоб дати просте визначення, REST – це будь-який інтерфейс між системами, що використовують HTTP для отримання даних та генерації операцій з цими даними у всіх можливих форматах, таких як XML та JSON. Це дедалі популярніша альтернатива іншим стандартним протоколам обміну даними, таким як SOAP (Simple Object Access Protocol), які мають високу ємність, але також є дуже складними. Іноді переважно використовувати більш просте рішення для обробки даних, таке як REST.

Переваги REST для розробки:

1. Поділ між клієнтом та сервером: протокол REST повністю відокремлює користувальницький інтерфейс від сервера та сховища даних. Це має деякі переваги при розробці додатків. Наприклад, це покращує портативність інтерфейсу на інших типах платформ, збільшує масштабованість проектів і дозволяє самостійно розвивати різні компоненти розробок.

2. Видимість, надійність та масштабованість. Поділ між клієнтом і сервером має одну очевидну перевагу, кожна команда розробників може масштабувати продукт без особливих проблем. Вони можуть мігрувати на інші сервери або вносити всі види змін у базу даних, за умови, що дані кожного запиту надсилаються правильно. Розділення полегшує розміщення спереду та ззаду на різних серверах, а це робить додатки більш гнучкими для роботи.

3. REST API завжди не залежить від типу платформи або мов: REST API завжди адаптується до типу синтаксису або платформ, що використовуються, що дає значну свободу при зміні або тестуванні нового середовища в рамках розробки. За допомогою REST API ви можете мати сервери PHP, Java, Python або Node.js. Єдине, що обов'язково, щоб відповіді на запити завжди надходили мовою, що використовується для обміну інформацією, як правило, XML або JSON.

Висновки. Так як сучасний світ взяв напрям на діджиталізацію, факт того, що освітній процес буде все далі переходити в електронний формат є невідворотнім. Варіант з описаним стартапом відцифровки гуртожитку може надати широкий спектр можливостей для його реалізації. Обрані технології дуже просто масштабуються та

легко підтримуються або взаємозаміняються в ході нових версій. Даний додаток може стати основою для зародження нового функціоналу платформи. Наша платформа допоможе тисячам студентів правильно організувати свій час і власний навчальний простір у гуртожитку, а працівникам вищого навчального закладу уникнути кілометрових черг і проблематичних ситуацій. Вся проблематика описана у даній статті не є безпідставною, вона тягнеться роками і саме створення нашого додатку може стати переломним моментом у відношенні до організації студентського життя.

Список використаної літератури

1. The Java EE 5 Tutorial [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/javaee/5/tutorial/doc/docinfo.html>.
2. Spring Framework Documentation [Електронний ресурс]. – Режим доступу: <https://docs.spring.io/spring/docs/current/spring-framework-reference/>
3. Hibernate ORM. Documentation – 5.4 [Електронний ресурс]. – Режим доступу: <https://hibernate.org/orm/documentation/5.4/>
4. Фаулер М. Шаблоны корпоративных приложений / М. Фаулер. – К.: Диалектика, 2019. – 544 с.
5. Хо К. Spring 5 для профессионалов / К. Хо, Р. Харроп, К. Шефер. . – 5-е изд. – К.: Вильямс, 2016. – 1120 с.
6. Browse MySQL Documentation by: Product [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/>
7. Top Reasons to Use MySQL [Електронний ресурс]. – Режим доступу: <http://www.databasequest.com/index.php/product-service/mysql-dbquest/>
8. REST API: What is it, and what are its advantages in project development? [Електронний ресурс]. – Режим доступу: <https://www.bbvaapimarket.com/en/api-world/rest-api-what-it-and-what-are-its-advantages-project-development/>
9. 8 Proven Reasons You Need Angular for Your Next Development Project [Електронний ресурс]. – Режим доступу: <https://www.grazitti.com/blog/8-proven-reasons-you-need-angular-for-your-next-development-project/>
10. Філдинг Р. Т. Архітектурні стилі та проектування мережеских архітектур програмного забезпечення / Р. Т. Філдинг, Р. Н. Тейлор, М. С. Акерман, Д. С. Розенблюм. – К.: УНІВЕРСИТЕТ КАЛІФОРНІЇ, 2000. – 180 с.