

Нікітенко Є.В.

Національний університет біоресурсів та природокористування України

Гуйда О.Г.

Таврійський національний університет імені В.І. Вернадського

ІНФОРМАЦІЙНА СИСТЕМА МОДЕЛЮВАННЯ ОСНОВНОГО ЗАКОНУ ДИНАМІКИ ОБЕРТАЛЬНОГО РУХУ

Авторами розроблено інформаційну систему моделювання фізичних законів на прикладі віртуальної лабораторної роботи для вивчення основного закону динаміки обертального руху. Комп'ютерне моделювання різних фізичних експериментів давно стало невід'ємною реальністю віртуального освітнього середовища. Віртуальні демонстрації та комп'ютерне моделювання фізичних процесів та явищ дуже поширені на ринку програмних продуктів. Ці системи мають загальну назву – віртуальна лабораторна робота, і є гідною альтернативою реальним лабораторним практикумам. На сьогодні є великий попит на такі програмні продукти. Найбільш активним споживачем є середня школа, а у вищих навчальних закладах така практика поступово інтегрується у навчальний процес, особливо там, де фізика не є базовою дисципліною для підготовки бакалаврів.

Актуальність роботи обумовлюється тим, що у зв'язку з постійною комп'ютеризацією суспільства та з погіршенням або дефіцитом матеріального забезпечення існує попит серед вищих навчальних закладів на комп'ютерні системи моделювання фізичних законів, а саме віртуальні лабораторні роботи, що допомагають студентам краще засвоїти складний матеріал, можуть мати багатий набір варіантів завдань та дозволяють проводити досліди з пристроями, які є небезпечними для життя, і моделювати умови, які неможливо створити у реальних лабораторіях.

У якості фізичної задачі було обрано основний закон динаміки обертального руху, теоретичний матеріал якого часто складно засвоюється студентами, особливо без проведення лабораторних робіт з реальними приладами.

Ключові слова: комп'ютерна система, моделювання фізичних законів, віртуальна лабораторна робота, закон динаміки обертального руху, фізична модель, OSGi, Java, JavaScript, контент.

Постановка проблеми. Однією з необхідних умов для успішного засвоєння навчального матеріалу студентами є проведення практичних занять. Це, в першу чергу, лабораторні роботи, на яких можна практично протестувати різні хімічні, фізичні та інші закони природи. На жаль, не завжди у навчальних закладах є можливість проводити експерименти на практиці. В результаті існують ситуації в навчальних закладах, де не вистачає, наприклад, хімікатів, і багато експериментів потребують дорогого обладнання. Крім того, часто лабораторне обладнання в школах та університетах просто застаріле або в дефіциті і сам процес дослідження вимагає значного часу і не викликає у студентів інтересу.

Однак сучасні інформаційні технології дозволяють частково компенсувати такі недоліки за допомогою спеціальних комп'ютерних систем – віртуальних лабораторних робіт. Це спеціальні комп'ютерні програми, що забезпечують моделювання різних процесів на комп'ютерах.

Подібні віртуальні лабораторні роботи мають безліч переваг для впровадження у навчальний процес та є альтернативою до реальних лабораторних занять.

Переваги віртуальних лабораторних робіт [1]:

1. Віртуальні лабораторні роботи надають можливість впровадження візуалізації фізичного закону на прикладі віртуального експерименту у лекцію або практичне заняття без підготовки спеціального обладнання.

2. Використання віртуальних лабораторних робіт дозволить навчати на достойному рівні навіть з низьким рівнем матеріального забезпечення навчального закладу.

3. Надається можливість проводити експерименти, які є небезпечними при проведенні в реальних умовах.

4. Надається можливість провести експеримент з критичними параметрами, що неможливо за реальних умов.

5. Проводити експеримент може кожен, хто має комп'ютер, без ризиків для життя та працездатності обладнання.

При дистанційній формі навчання, коли студенти не мають можливості проводити реальні лабораторні роботи, віртуальні лабораторні роботи є вкрай необхідні. Вони можуть бути представлені як у вигляді веб-додатків, так і у вигляді настільних програм.

Формулювання цілей статті

Метою статті є створення комп'ютерної системи моделювання основного закону динаміки обертального руху у вигляді віртуальної лабораторної роботи.

Для досягнення цієї цілі було поставлено такі завдання:

- вивчити існуючі аналоги інформаційних комп'ютерних систем, що візуалізують фізичні експерименти та проаналізувати їх недоліки та переваги;
- розробити вимоги та технічне завдання для комп'ютерної системи, яка покриє недоліки аналогів та буде здатна до розширення та підтримки;
- спроектувати систему та допоміжні модулі;
- створити програмну частину – віртуальну лабораторну роботу, як модуль системи для вивчення основного закону динаміки обертального руху у вигляді веб-додатку;
- зробити тестування системи.

Теоретичною основою дослідження стали концепція побудови комп'ютерних систем з використанням специфікації OSGi, документація існуючих бібліотек для моделювання фізичних експериментів та положення щодо архітектури багатомодульного програмного додатку.

Виклад основного матеріалу. Проаналізувавши доступні аналоги віртуальних лабораторних робіт, були створені вимоги для майбутньої системи. При цьому були сформовані функціональні і системні вимоги та вимоги до інтерфейсу. До функціональних вимог системи можна віднести:

- деталізоване моделювання фізичних процесів;
- володіння можливостями до розширення контенту та функціоналу;
- можливість виконання лабораторної роботи за шаблоном або шаблонами, представленими до кожного експерименту;
- позбавлення користувачів самостійних складних та великих розрахунків, надавши до шаблону лабораторної спеціалізовані калькулятори;
- надання можливості вибрати переклад опису експерименту та шаблонів лабораторних робіт міжнародною мовою;
- система повинна детально та поступово симулювати експеримент зі зрозумілими підказками.

Разом з функціональними вимогами були сформовані системні вимоги:

- архітектура додатку повинна бути реалізована з використанням специфікації OSGi і сервісної платформи для Java-додатків;
- система повинна містити модуль моделювання фізичних експериментів, які написані мовою програмування JavaScript;
- система повинна мати можливості для керування контентом, що полегшить організацію, управління та доставку авторського контенту;
- система повинна бути реалізована у вигляді зручного веб-додатка;
- система повинна бути спроможною інтегруватися, як додаток, або, як допоміжний ресурс, для систем дистанційного навчання вищих навчальних закладів.

Також були розроблені вимоги до інтерфейсу системи. Інтерфейс повинен містити:

- компонент з моделюванням фізичного експерименту;
- компонент для конфігурації експериментальної установки;
- компонент з шаблонами лабораторних робіт, що містять теоретичні відомості, опис установки, порядок та хід проведення;
- компонент с допоміжними калькуляторами для спеціалізованих розрахунків;
- форму для збереження проміжних результатів експерименту;
- компонент для відправки отриманих результатів на перевірку;
- компонент з підказками.

Враховуючи всі вимоги, які були поставлені до системи, було вирішено використовувати дворівневу архітектуру «клієнт-сервер», яка розділяє завдання або навантаження між постачальниками ресурсу або сервісу, що називаються серверами, та споживачами служб, які називаються клієнтами. Клієнт не надає ніякого ресурсу, але запитує контент або службову функцію у сервера. Сервери та клієнти мають окреме програмне забезпечення. Клієнти починають спілкування з серверами, які чекають на вхідні запити.

Клієнтська сторона – це сторінка сайту у веб-браузері з віртуальною лабораторною роботою, а сервер – це окреме програмне забезпечення, що має свої сервіси для контенту. Зрозуміло, що сервер і клієнт відповідають за різні функції, і тому для їх надання існує зовсім різний стек технологій. Тому слід розглянути вибір технології для кожної з цих частин окремо.

Серверна частина

Серверна частина відповідає за збереження, модифікацію, відновлення та керівництво даними. Специфікація OSGi була вибрана через те, що її мета задовольняє наші потреби, адже вона дає модель для побудови програм з компонентів, пов'язаних разом за допомогою сервісів. Суть полягає в можливості переінсталяції динамічних компонентів і складових частин програми без необхідності її зупинки і перезапуску.

Плагіни платформи OSGi містять Java-класи та інші ресурси, які разом можуть реалізовувати деякі функції, а також надавати послуги і пакети іншим плагінам. Конструктивно, плагін може бути або каталогом, або jar-архівом. Основний спосіб взаємодії між плагінами здійснюється за допомогою сервісів (об'єктів, які зареєстровані в ядрі системи з заявленими реалізованими інтерфейсами). Плагіни реєструють сервіси для надання певної функціональності іншим плагінам. Крім цього, фреймворк OSGi надає механізм створення і обробки подій, управління імпортом/експортом java-пакетів і бібліотек, набір класлоадерів, методи адресації ресурсів [2].

В якості прикладів тематики плагінів можна виділити наступні:

- UI-шаблони, UI-компоненти, сторінки, стилі, скрипти;
- контролери для взаємодії клієнта з сервером;
- сервіси, провайдери та інші утилітарні класи для обслуговування запитів всередині сервера.

Основною метою роботи є створення програмного продукту, що інтегрується у навчальний процес, тому вибір фреймворку OSGi є цілком доречним, оскільки, завдяки його можливостям, можна без перезапуску додатку встановлювати, підключати, відключати і оновлювати модулі системи. Це означає, що не потрібно зупиняти сервер, встановлювати нову версію програми, перевіряти та запускати сервер [3].

Під час проектування серверної частини комп'ютерної системи було запропоновано спроектувати прототип системи з використанням особливостей систем управління контентом. Адже тоді, вимога щодо можливості управління контентом задовольняє відразу декілька інших вимог.

Система керування вмістом (СКВ) – інформаційна система або комп'ютерна програма, яка використовується для забезпечення і організації спільного процесу створення, редагування і управління контентом.

Основними функціями СКВ є:

- надання інструментів для створення вмісту, організація спільної роботи над контентом;

- управління контентом: зберігання, контроль версій, дотримання режиму доступу, управління потоком документів;

- публікація контенту;
- подання інформації у вигляді, зручному для навігації та пошуку.

В системі керування вмістом можуть перебувати найрізноманітніші дані: документи, фільми, фотографії, номери телефонів, наукові дані і т.і. Така система часто використовується для зберігання, управління, перегляду і публікації документації. Контроль версій є одним з основних її переваг, коли вміст змінюється групою осіб.

Більшість сучасних СКВ мають модульну архітектуру, що дозволяє адміністратору самому вибирати і налаштовувати ті компоненти, які йому необхідні. В нашій системі можна виділити такі компоненти:

- компонент з візуалізацією фізичного експерименту;
- панель для шаблонів лабораторних робіт;
- шаблон лабораторної роботи;
- спеціалізовані калькулятори;
- форми та таблиці для інтерактивного виконання та заповнення результатів.

Сайти, які організовані за допомогою системи керування вмістом, базуються на наступних технологіях: веб-сервер, сховище даних (зазвичай СУБД, наприклад, такі, як MySQL або PostgreSQL, але існують і noSQL), веб-додаток для забезпечення роботи самої системи, візуальний редактор сторінок, файловий менеджер з веб-інтерфейсом для управління файлами сайту, система управління правами користувачів і редакторів сайту.

Існує термін «контент-менеджер», який означає вид професійної діяльності редактора сайту або співробітника, який працює з СКВ. Так як планується інтеграція з системами дистанційного навчання, то таку роль можуть отримати викладачі. За допомогою таких привілеїв вони зможуть самостійно створювати сторінки з новими реалізаціями моделювання фізичних процесів, додавати авторські шаблони для виконання лабораторного практикуму та створювати спеціалізовані калькулятори.

Головною метою створення на серверній стороні системи керування вмістом полягає в тому, щоб створити для викладачів платформу для створення власних прикладів лабораторних практикумів з існуючих компонентів, що можуть конфігуруватися. Тобто, якщо будуть існувати компоненти, які моделюють установку та експерименти, компоненти опису лабораторного практикуму, компоненти спеціалізованих калькуляторів та

різноманітні шаблони форм, таблиць для збереження результатів, то автор зможе створити власну віртуальну лабораторну роботу. Більш того, якщо автор володітиме спеціальними правами, то матиме змогу самостійно створювати компоненти та шаблони, щоб розвивати, удосконалювати систему та покривати потреби у необхідних компонентах.

Клієнтська частина

Клієнтська частина – це те, що бачить користувач. Так як задовольнити потреби кожного користувача у зручності інтерфейсу дуже важко, було запропоновано приблизити вигляд веб-додатку до аналогів. Тобто користувач не матиме необхідності у вивченні нового інтерфейсу, а буде орієнтуватися завдяки досвіду користувача.

Майже всі симуляції фізичних експериментів у аналогів виконані з використанням flash-анімацій. На сьогоднішній день розробники сайтів намагаються уникати flash-анімацій, бо вони мають низку недоліків:

- технологія flash-анімації вимагає додаткових системних ресурсів, тому існує велика ймовірність втрати клієнтів з нешвидким Інтернет-каналом і комп'ютерами невисокої продуктивності;
- у відвідувачів не завжди є можливість запустити flash-додаток, іноді необхідно встановити плагін або оновити його до необхідної версії;
- деякі браузерери потребують дозвіл на використання flash-анімацій.

На заміну flash-анімаціям фізичних експериментів запропоновано використання іншої технології для створення візуальних ефектів. В нашому випадку це мова програмування JavaScript, яка буде вимальовувати 2D-експеримент на веб-сторінці на такому елементі HTML, як canvas-полотно. Написання програмного коду для скрипта на JavaScript для візуалізації фізичної установки є нетривіальною задачею. Тому це створює незручності, оскільки для того, щоб симулювати фізичний експеримент, треба дотримуватися законів и запрограмувати відповідні правила.

Щоб уникнути цього була проаналізована наявність безкоштовних JavaScript-бібліотек. Зазвичай ці бібліотеки мають вбудовані фізичні закони. До таких бібліотек відносяться наступні [4][5][6]:

- Matter.js;
- Planck.js;
- P5.js;
- Physics.js;
- Vox2D.js.

В основному ці бібліотеки розраховані на створення веб-браузерних ігор та розважального контенту. Використання цих інструментів для

створення скриптів для моделювання фізичних процесів є більш доречним. Але всі мають свої обмеження у вигляді малої бази написаного коду та складності документації. Кожна з розглянутих бібліотек була не в змозі при дослідженні основного закону динаміки обертального руху надати можливості для програмування тросу, який перекинутий через нерухомий вал. Також фреймворки не надають можливості для реалізації накладених компонентів, а саме хрестовини у маятника.

Для реалізації інших компонентів, необхідних для віртуальної лабораторної роботи, краще за все підходить нова версія мови розмітки HTML5. Ця технологія описує та визначає вміст веб-сторінки разом із основним макетом веб-сторінки. Головною метою використання цієї технології у рамках системи є розподіл складових віртуальної лабораторної роботи на семантичні компоненти, які будуть точно та конкретно описувати те, що представляє собою даний контент.

Також для презентації та відображення веб-сторінки підходить формальна мова опису зовнішнього вигляду сторінки CSS. CSS застосовується до документів, що написані за допомогою мови розмітки, тому вибір цих технологій є цілком обумовленим.

Діаграми прецедентів та класів

При проектуванні системи були розроблені діаграми прецедентів та класів. Діаграма прецедентів (рис. 1) показує взаємозв'язок між користувачем та різними випадками використання, в яких користувач бере участь.

Мета діаграми прецедентів полягає в тому, щоб забезпечити високий рівень перегляду системи та передати вимоги, викладені на загальних умовах.

Під час проектування клієнтської частини були виявлені наступні класи-кандидати:

1. OberbekWheel, який представляє фізичну модель маятника Обербека;
2. OberbekWheelView, який відповідає за графічне зображення та анімацію маятника Обербека;
3. DrawCanvas2DService – утилітарний сервіс, що взаємодіє з canvas-елементом у веб-документі та відповідає за зображення примітивних фігур.

Ці класи були розроблені за допомогою патерна проектування MVC (Model – View – Controller). На основі класів-кандидатів була створена діаграма класів (рис. 2).

Реалізація комп'ютерної системи

Концепція показу контенту для клієнту полягає у відображенні набору готових компонентів відповідно до шаблону. Це означає, що контент-автор наповнює шаблон веб-сторінки необхідними компонентами, який потім стає суцільною веб-

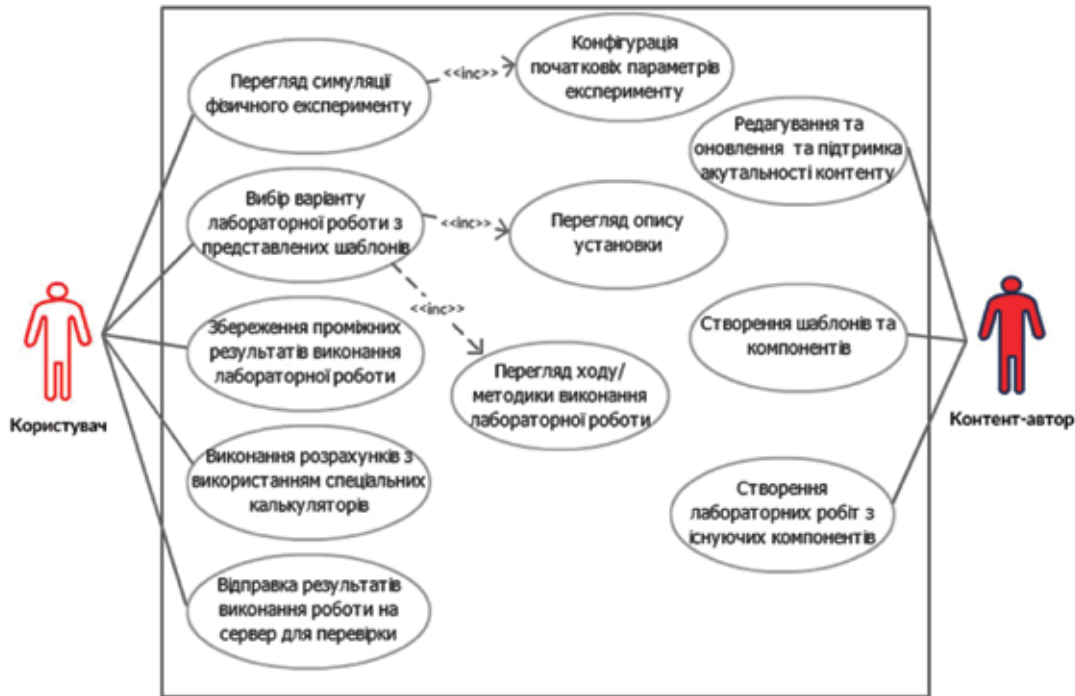


Рис. 1. Діаграма прецедентів

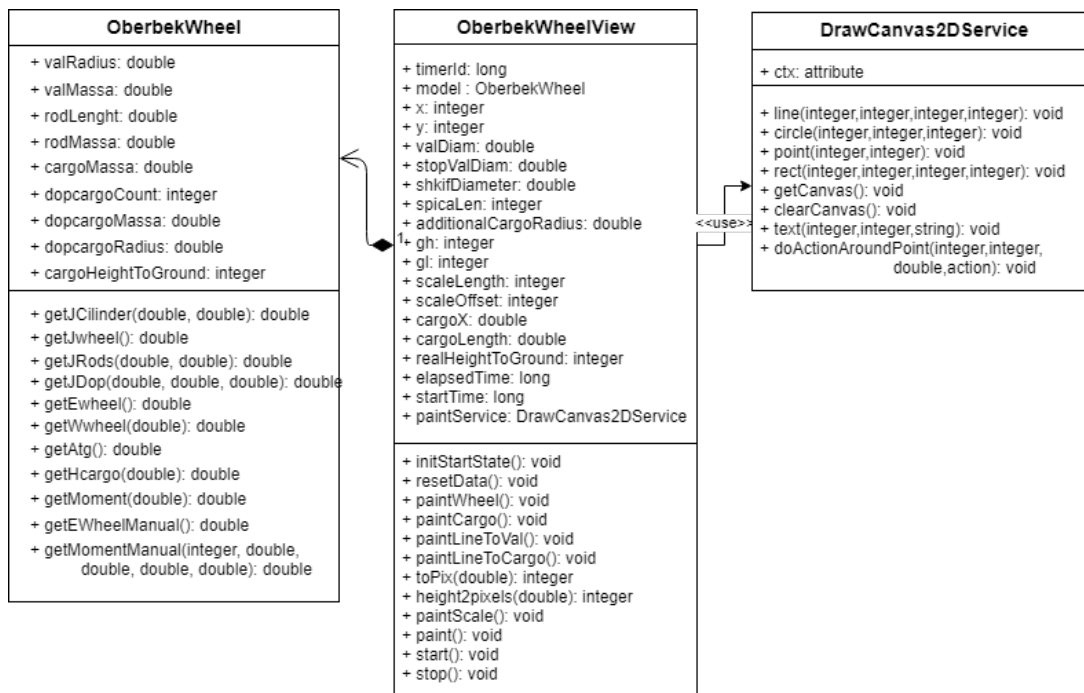


Рис. 2. Діаграма класів

сторінкою з готовою для виконання віртуальною лабораторною роботою.

Так як ключовими одиницями клієнтської програми є шаблон та компонент, то процес реалізації клієнтського програмного забезпечення було розділено на декілька частин:

1. Створення шаблону структури віртуальної лабораторної роботи;

2. Реалізація компонентів, необхідних для віртуальної лабораторної роботи.

Для створення шаблону були проаналізовані такі компоненти:

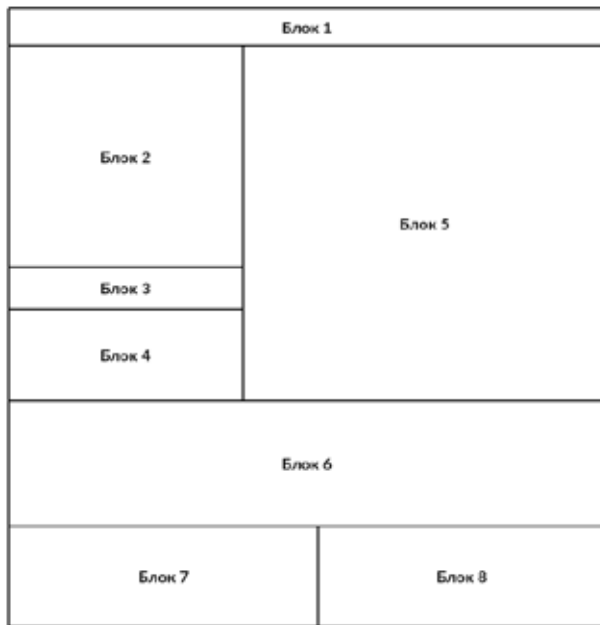


Рис. 3. Шаблон віртуальної лабораторної роботи

1. Компонент візуалізації фізичного експерименту;
2. Компонент з варіантами лабораторних робіт;
3. Компонент зі спеціальними калькуляторами;
4. Компонент з таблицею отриманих результатів;
5. Компонент з формою відправки результатів на сервер.

Базуючись на аналогах, переліку компонентів та їх семантичному значенні для віртуальної лабораторної роботи був розроблений шаблон (рис. 3).

Шаблон розділений на блоки, які призначені для конкретного компоненту:

1. Блок 1 – для компоненту, що містить тему та категорії лабораторної роботи, ім'я автора.
2. Блок 2 – для компоненту, що моделює фізичний експеримент.
3. Блок 3 – для компоненту, що контролює симуляцію фізичного експерименту.
4. Блок 4 – для компоненту, що представляє різноманітні конфігурації обладнання для фізичного експерименту.
5. Блок 5 – для компоненту, що містить варіанти лабораторних робіт з описом фізичної установки, теоретичним матеріалом, методичними вказівками та ходом роботи.
6. Блок 6 – для компоненту, що містить таблицю результатів, отриманих під час проведення експерименту.
7. Блок 7 – для компоненту, що представляє спеціалізовані до лабораторної роботи калькулятори.
8. Блок 8 – для компоненту, що містить форму для відправки отриманих результатів на сервер або для відображення службової інформації.

Шаблон містить мінімально необхідний набір компонентів, які повинні брати участь у віртуальній лабораторній роботі, тому він здатний підлаштуватися під різні типи віртуальних лабораторних робіт. Однією з головних властивостей розробленого шаблону є те, що за відсутності одного з блоків, структура шаблону залишається незмінною, а значить відображення теж не зазнає змін.

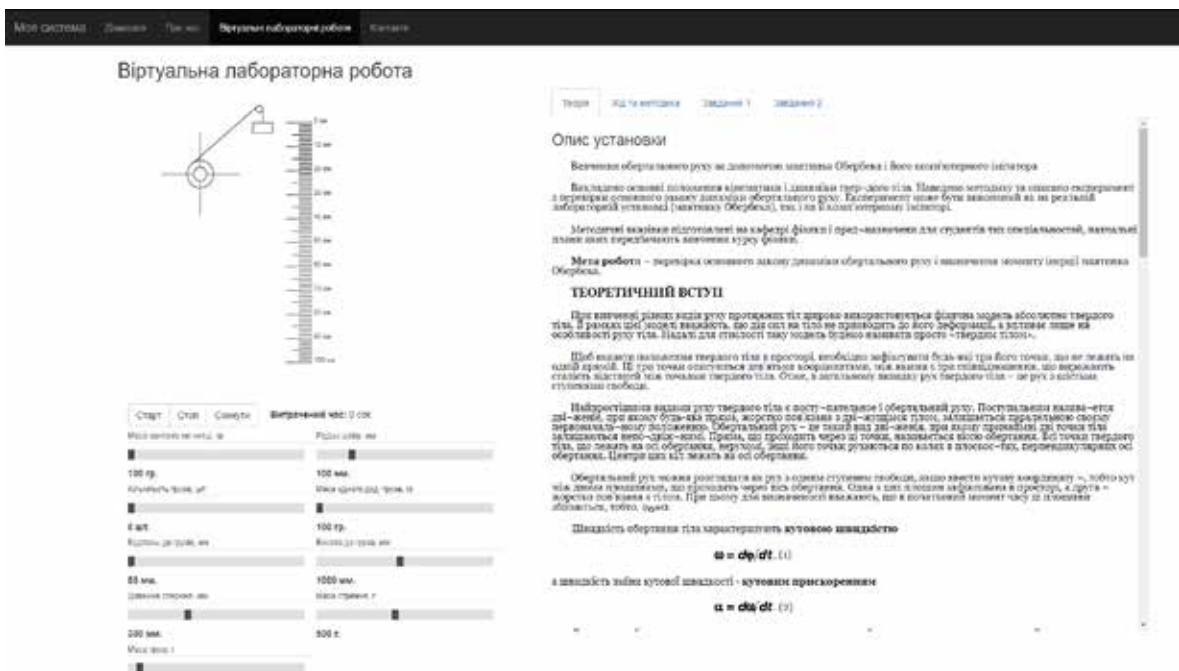


Рис. 4. Зовнішній вигляд частини підготовленої веб-сторінки

Таблиця результатів експерименту

№ пп	m, кг	r, м	h, м	t, сек	t _{сп} , сек	M, Н * м	ε, рад/с ²	
1	0.1	0.01	4.48	4.48	4.48	0.0965	1.0067	--
				4.481				
				4.48				
2	0.5	0.01	1	4.52	4.521	0.0971	0.9859	--
				4.521				
				4.521				
3	0.1	0.02	1	1	2.321	0.1885	1.9235	--
				2.321				
				2.32				
4	0.5	0.02	1	2.4	2.4	0.1890	1.7839	--
				2.41				
				2.4				

Калькулятори

Час	Обчислити		Кутове прискорення
Час	Обчислити		Момент сили
Обчислити			Момент інерції

Форма для відправки результатів

Ім'я:

Емейл:

Деталі:

Комп'ютерна система моделювання фізичних процесів

Рис. 5. Зовнішній вигляд частини підготовленої веб-сторінки

Веб-сторінка віртуальної лабораторної роботи

Після реалізації усіх необхідних компонентів на основі попереднього шаблону була створена веб-сторінка (рис. 4, рис. 5).

Висновки. В роботі були сформовані функціональні і системні вимоги та вимоги до інтерфейсу. Для деталізації варіантів використання програми була створена діаграма прецедентів, а для зручної реалізації – діаграма класів.

Були розглянуті найбільш доречні інструменти для реалізації системи як для серверної частини, так і для клієнтської. Для серверної частини було вибрано OSGi-технологію з мовою програмування Java. Для клієнтської частини було вибрано стек тех-

нологій HTML5, CSS з використанням бібліотеки Bootstrap та JavaScript. Із-за відсутності готових JavaScript 2D-фізичних рушіїв, які могли б візуалізувати фізичний процес, було вирішено реалізувати моделювання з використанням фізичної моделі, створеної на основі аналітичних формул.

Було створено клієнтську частину комп'ютерної системи моделювання закону динаміки обертового руху, яка представлена у вигляді працюючої веб-сторінки з віртуальною лабораторною роботою.

Процес тестування клієнтської частини показав працездатність розробленої системи. Шляхом виконання розрахунків наведених у лабораторній роботі було визначено, що фізична модель відповідає дійсності.

Список літератури:

1. Гергова И.Ж., Коцева М.А., Ципинова А.Х., Шериева Э.Х., Азизов И.К. Виртуальные лабораторные работы как форма самостоятельной работы студентов [Електронний ресурс]// Современные наукоемкие технологии. – 2017. – № 1. – с. 94-98; URL: <http://www.top-technologies.ru/ru/article/view?id=36562>
2. Sunil Patil. Hello, OSGi, Part 1: Bundles for beginners [Електронний ресурс]// Creating, executing, and managing bundles in an OSGi container, 2008-2018. URL: <https://www.javaworld.com/article/2077837/application-development/java-se-hello-osgi-part-1-bundles-for-beginners.html>
3. Павел Самолисов. Введение в OSGi.[Електронний ресурс]// Знакомимся с "серебряной пулей" для разработки модульных приложений, 2009-2018. URL: <http://samolisov.blogspot.com/2009/03/osgi.html>
4. Liam Liabru. Git Hub Repository. [Електронний ресурс]// Matter.js, 2017-2018. URL: <https://github.com/liabru/matter-js/wiki>
5. Ali Shakiba. Git Hub Repository. [Електронний ресурс]// Planck.js, 2018. URL: <https://github.com/shakiba/planck.js>
6. Processing Foundation. Processing. [Електронний ресурс]// p5.js as a JS client-side library for creating graphic and interactive experiences, based on the core principles of Processing, 2017-2018. URL: <https://p5js.org/reference/>

Nikitenko Ye.V., Guida O.G. INFORMATION SYSTEM OF SIMULATION OF THE BASIC LAW OF THE DYNAMICS OF ROTATIONAL MOTION

The authors developed an information system for modeling physical laws using the example of virtual laboratory work to study the basic law of rotational motion dynamics. Computer modeling of various physical experiments has long become an integral reality of the virtual educational environment. Virtual demonstrations and computer simulations of physical processes and phenomena are very common in the market of software products. These systems have a common name - virtual laboratory work, and are a worthy alternative to real laboratory practices. Today there is a great demand for such software products. The most active consumer is the secondary school, and in higher educational institutions such practice is gradually integrated into the educational process, especially where physics is not a basic discipline for the preparation of bachelors.

The relevance of the work is determined by the fact that in connection with the constant computerization of society and with the deterioration or shortage of material support, there is a demand among higher educational institutions for computer systems for modeling physical laws, namely virtual laboratory work that helps students better learn complex material, can have a rich set of task options and allow experiments with devices that are life-threatening and simulate conditions that cannot be created in real laboratories.

As a physical problem, the basic law of the dynamics of rotary motion was chosen, the theoretical material of which is often difficult for students to learn, especially without conducting laboratory work with real devices.

Key words: *computer system, simulation of physical laws, virtual laboratory work, law of dynamics of rotary motion, physical model, OSGi, Java, JavaScript, content.*