**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

Кафедра обчислювальної техніки та програмування

**ЗВІТ**

з переддипломної практики

Спеціальність - 123 Комп’ютерна інженерія

Строки проведення практики: 02.09.2020р. – 25.10.2020р.

**Виконала:**

студ. групи КІТ-М119б

Щербініна Є.М. \_\_\_\_\_\_\_\_\_\_\_

(підпис)

**Керівники практики:**

від університету:

Філоненко А.М. \_\_\_\_\_\_\_\_\_\_\_

(підпис)

Харків 2020

ЗМІСТ

[ВСТУП 3](#_Toc54387693)

[1. ШИФРУВАННЯ 4](#_Toc54387694)

[2. ПОНЯТТЯ ВЕБ-СЕРВІСУ ХМАРНОГО ЗБЕРІГАННЯ ФАЙЛІВ 5](#_Toc54387695)

[3. ВИБІР МОВИ ПРОГРАМУВАННЯ ДЛЯ СЕРВЕРНОЇ ЧАСТИНИ 7](#_Toc54387696)

[4. ОГЛЯД МОВИ ПРОГРАМУВАННЯ JAVA 8](#_Toc54387697)

[5. ВИБІР ТЕХНОЛОГІЙ ДЛЯ ПРОЕКТУВАННЯ БАЗИ ДАНИХ 12](#_Toc54387698)

[6. ВИБІР ТЕХНОЛОГІЇ СТВОРЕННЯ ВЕБ-САЙТІВ 14](#_Toc54387699)

[7. ОГЛЯД ТЕХНОЛОГІЇ JSP 16](#_Toc54387700)

[8. ОГЛЯД ТЕХНОЛОГІЇ DAO 19](#_Toc54387701)

[ВИСНОВОК 19](#_Toc54387702)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 20](#_Toc54387703)

ВСТУП

Метою переддипломної практики є огляд використовуваних методів та технологій для проектування веб-сервісу хмарного зберігання файлів. Огляд необхідності шифрування.

Основні задачі проведення переддипломної практики:

* ознайомлення зі структурою підприємства бази практики в цілому та підрозділів, де проходження практики має місце;
* вивчення теоретичних питань, які складають основу задачі, практичне вирішення якої передбачене у виданому студентові індивідуальному завданні;
* виконання пов’язаних з виробничою необхідністю конкретних практичних завдань з спеціальності та у відповідності з програмою практики.

1. ШИФРУВАННЯ

Безпека даних - найважливіше завдання в сучасному світі. Влада, компанії та інші організації втратили багато грошей, а багато інших закрилися, через діяльність хакерів і зловмисників. За минулі роки були розроблені різні схеми шифрування для захисту бази даних від атак зловмисників. Оскільки дані - це життєвий канал кожної організації, існує потреба в безпечному зберіганні даних, щодня генерованих цими організаціями. Хмарне сховище необхідно для віддаленого зберігання даних.

Для багатьох підприємств безпека даних є однією з основних проблем при відправці файлів в хмару. Вони турбуються про те, що їх файли будуть переглянуті або навіть скомпрометовані.

Для захисту цілісності даних, що зберігаються, необхідні методи шифрування даних. У минулому багато компаній відчували себе комфортно, дозволяючи постачальникам хмарних послуг управляти всіма своїми даними, вважаючи, що ризиками безпеки можна керувати за допомогою контрактів, засобів контролю і аудиту. Однак з часом стало очевидно, що постачальники хмарних послуг не можуть виконувати такі зобов'язання.

1. ПОНЯТТЯ ВЕБ-СЕРВІСУ ХМАРНОГО ЗБЕРІГАННЯ ФАЙЛІВ

Хмарні сервіси - це сервіси, що працюють на хмарних сховищах. Тобто, їх не потрібно встановлювати на комп'ютер і отримувати доступ з будь-якої точки виходу. В онлайн (хмарних) сховищах дані зберігаються на численних розподілених в мережі серверах, що надаються в користування клієнтам, в основному, третьою стороною. На відміну від моделі зберігання даних на власних виділених серверах, придбаних або орендованих спеціально для подібних цілей, кількість або будь-яка внутрішня структура серверів клієнту, в загальному випадку, не доступна.

Дані зберігаються і обробляються в так званій «хмарі», яка представляє собою, з точки зору клієнта, один великий віртуальний сервер. Фізично ж такі сервери можуть розташовуватися віддалено один від одного географічно. Серед найбільш відомих хмарних сервісів:

* Dropbox;
* Microsoft OneDrive;
* Google Drive;
* iCloud;
* Mega;
* Fex.net.

Хмарне сховище надається на вимогу в необхідному обсязі, оплачується за фактом використання і позбавляє від необхідності купувати власну інфраструктуру для зберігання даних і керувати нею. Це забезпечує гнучкість, глобальну масштабованість і надійність. Дані доступні в будь-який час і в будь-якому місці.

Додатки отримують доступ до хмарного сховища через традиційні протоколи зберігання даних або безпосередньо через API. Багато постачальників пропонують додаткові сервіси, призначені для захисту, збору і аналізу даних у величезних масштабах, а також управління ними [1].

Переваги хмарного зберігання файлів:

* можливість доступу до даних з будь-якого комп'ютера, що має вихід в Інтернет;
* можливість організації спільної роботи з даними;
* висока ймовірність збереження даних навіть у разі апаратних збоїв;
* клієнт платить тільки за те місце в сховищі, яке фактично використовує, але не за оренду сервера, всі ресурси якого він може і не використовувати;
* клієнту немає необхідності займатися придбанням, підтримкою і обслуговуванням власної інфраструктури зі зберігання даних, що, в кінцевому рахунку, зменшує загальні витрати виробництва;
* всі процедури по резервуванню і збереженню цілісності даних виробляються провайдером «хмарного» центру, яка не втягує в цей процес клієнта [2].

1. ВИБІР МОВИ ПРОГРАМУВАННЯ ДЛЯ СЕРВЕРНОЇ ЧАСТИНИ

Для опису логіки сайту і генерації HTML використовується одна з серверних мов програмування. Теоретично, для створення сайтів можна використовувати майже будь-яку мову, але так склалося, що тільки деякі з них популярні для веб проектування.

Серверну частину веб додатку можна написати за допомогою таких мов програмування, як - PHP, JavaScript, Ruby, Python, Erlang / Elixir, Go, Clojure, Java / Kotlin / Scala, C #. Хоча найбільш популярними серед них є PHP, JavaScript та Java. Обирати мову програмування я буду серед цих трьох. На рис. 1 зображено діаграму популярності приведених мов програмування.

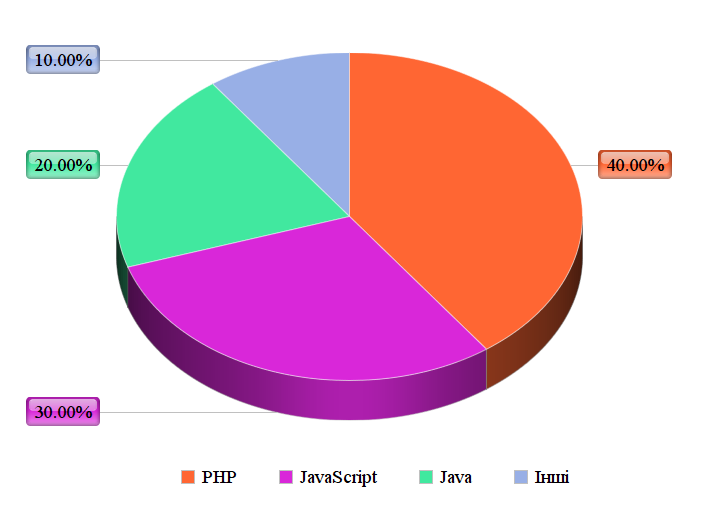


Рисунок 1 – Діаграма популярності мов програмування для серверної частини

Згідно діаграми можна зробити висновок, що найбільш популярною мовою для веб програмування є PHP, мови JavaScript та Java доволі сильно відстають. Логічним було б використання мови програмування PHP, проте її актуальність все ж таки починає поволі знижуватися. Хоча потрібно звернути увагу не лише на популярність. Враховуючи специфіку створюваного продукту, необхідно врахувати необхідність легкого підключення до бази даних та постійного звернення до неї, з метою збереження даних про файли будь-якого типу. Ідеальним варіантом є використання інтернет-протоколу JDBC, який можливий при виборі мови програмування Java.

У цілому використання мови програмування Java для серверної частини та допоміжної мови JavaScript для надання веб-сторінкам динамічності є ідеальним варіантом.

1. ОГЛЯД МОВИ ПРОГРАМУВАННЯ JAVA

За своїм призначенням мови програмування бувають:

* **спеціальні** (вирішують задачі у певній галузі. Наприклад, у математиці Fortran, в базах даних SQL);
* **загального призначення** (вирішують широкий спектр задач у будь-яких галузях. Наприклад, С++, Java, С#).

Java входить до другого типу - це мова загального значення, на якій можна написати практично будь-яку програму. Сьогодні на Java створюються програми будь-якого рівня складності: від програмування побутової техніки та мобільних приладів ([Java МЕ](https://ru.wikipedia.org/wiki/Java_Platform,_Micro_Edition" \t "_blank)) до складних високонавантажених сервісів ([Java ЕЕ](https://ru.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition" \t "_blank)).

У створенні мови програмування Java було п'ять початкових цілей:

* синтаксис мови повинен бути простим, об'єктно-орієнтовним та звичним;
* реалізація має бути безвідмовною та безпечною;
* повинна зберегтися незалежність від архітектури та переносність;
* висока продуктивність виконання;
* мова має бути інтерпретованою, багатонитевою, із динамічним зв'язуванням модулів.

Під «незалежністю від архітектури» мається на увазі те, що програма, написана на мові Java, працюватиме на будь-якій підтримуваній апаратній чи системній платформі без змін у початковому коді та перекомпіляції.

Цього можна досягти, компілюючи початковий Java код у байт-код, який є спрощеними машинними командами. Потім програму можна виконати на будь-якій платформі, що має встановлену віртуальну машину Java, яка інтерпретує байткод у код, пристосований до специфіки конкретної операційної системи і процесора. Зараз віртуальні машини Java існують для більшості процесорів і операційних систем [3].

Стандартні бібліотеки забезпечують загальний спосіб доступу до таких платформозалежних особливостей, як обробка графіки, багатопотоковість та робота з мережами. У деяких версіях задля збільшення продуктивності JVM байт-код можна компілювати у машинний код до або під час виконання програми.

Основна перевага використання байт-коду – це портативність. Тим не менш, додаткові витрати на інтерпретацію означають, що інтерпретовані програми будуть майже завжди працювати повільніше, ніж скомпільовані у машинний код. Проте, цей розрив суттєво скоротився після введення декількох методів оптимізації у сучасних реалізаціях JVM.

В Java можливе тільки одинарне успадкування, завдяки чому виключається можливість конфліктів між членами класу (методи і змінні), які успадковуються від базових класів.

У намірах проектувальників Java мала замінити C++ – об'єктного наступника мови C. Проектувальники почали з аналізу властивостей C++, які є причиною найбільшого числа помилок, щоби створити просту, безпечну і безвідмовну мову програмування.

В Java існує система винятків або ситуацій, коли програма зустрічається з неочікуваними труднощами, наприклад:

* операції над елементом масиву поза його межами або над порожнім елементом;
* читання з недоступного каталогу або неправильної адреси URL;
* ввід недопустимих даних користувачем.

Одна з особливостей концепції віртуальної машини полягає в тому, що помилки (виключення) не призводять до повного краху системи. Крім того, існують інструменти, які «приєднуються» до середовища періоду виконання і кожен раз, коли сталося певне виключення, записують інформацію в пам'яті для налагодження програми. Ці інструменти автоматизованої обробки виключень надають основну інформацію щодо виключень в програмах на Java [4].

Java - це не тільки мова програмування, а й екосистема інструментів, що охоплює майже все, що може знадобитися при програмуванні на Java. У неї входять:

* Java Development Kit (JDK) - комплект розробника Java. За допомогою JDK і стандартного блокнота можна писати і запускати / компілювати код на Java;
* Java Runtime Environment (JRE) - виконуюча система Java. Механізм розповсюдження програмного забезпечення, складається з автономної віртуальної машини Java, стандартної бібліотеки Java (Java Class Library) і інструментів настройки;
* Integrated Development Environment (IDE) - інтегроване середовище розробки. Інструменти, які допомагають запускати, редагувати і компілювати код. Найпопулярніші з них - IntelliJ IDEA, Eclipse і NetBeans.

Java можна знайти всюди. Це основна мова розробки для Android. Він використовується в веб-додатках, урядових веб-сайтах і технологіях обробки великих даних, таких як Hadoop і Apache Storm. Java підходить і для наукових проектів, особливо в області обробки природної мови.

Незважаючи на зростаючу популярність Go і Python, Java залишається на вершині списку вже понад десятиліття.

Як Java змінила світ програмування:

* гнучкість. Java довела, що C - процедурний, керований вручну і залежний від платформи код - це не межа досконалості. Завдяки Java, все більше людей почали застосовувати об'єктно-орієнтоване програмування, яке зараз використовується повсюди;
* аплети. Ще до появи JavaScript, в Java додали аплети - невеликі веб-програми, які надають інтерактивні елементи для візуалізації та навчання. Вони не використовуються ні для чого, крім простої анімації, однак аплети привернули увагу багатьох програмістів і підштовхнули їх до розробки HTML5, Flash і JavaScript;
* розробка через тестування – вже давно не експериментальна практика, а стандартний спосіб розробки програмного забезпечення. Введення JUnit 2000 року вважається одним з найбільших досягнень Java.

Java – вже не єдина офіційно підтримувана мова для розробки на Android. Java далеко не єдиний вибір в веб-програмуванні. Проте, Java йде в ногу з часом. Давайте розглянемо, які переваги пропонує Java:

* Java включає в себе об'єктно-орієнтоване програмування, що дозволяє повторно використовувати об’єкти в інших програмах, більш ефективно організовувати структуру програми, а також спрощує обслуговування та модернізацію старого коду;
* це мова високого рівня, що значно спрощує розробку;
* Java підтримує безліч бібліотек, що дозволяє розробникам створювати будь-які функції;
* безпечність;
* незалежність від платформи («Написати один раз та використовувати будь-де»);
* Java має вбудований механізм спільного використання даних і програм декількома комп'ютерами, що підвищує продуктивність і ефективність праці;
* автоматичне керування пам’яттю;
* багатопоточність;
* стабільність та постійний розвиток [5].

1. ВИБІР ТЕХНОЛОГІЙ ДЛЯ ПРОЕКТУВАННЯ БАЗИ ДАНИХ

База даних – це сукупність певним чином організованих даних, що зберігаються в запам'ятовуючих пристроях ЕОМ. Зазвичай дані зберігаються на жорсткому диску сервера організації.

У загальному випадку дані в базі даних (в великих системах) є інтегрованими і розділяються. Ці два аспекти, інтеграція і поділ даних, являють собою найбільш важливі переваги використання баз даних на «великому» обладнанні і, щонайменше, один з них – інтеграція – є перевагою їх застосування і на «малому» обладнанні.

Під поняттям інтеграції даних мається на увазі можливість представити БД як об'єднання декількох окремих файлів даних, повністю або частково виключаючи надмірність зберігання інформації.

Під поняттям розділу даних мається на увазі можливість використання декількома різними користувачами окремих елементів, що зберігаються в базі даних. Мається на увазі, що кожен з користувачів зможе отримати доступ до одних і тих же даних, можливо, навіть одночасно (паралельний доступ). Такий поділ даних, з паралельним або послідовним доступом, частково є наслідком того факту, що БД має інтегровану структуру.

Одним із наслідків згаданих вище характеристик бази даних (інтеграції та роздільності) є те, що кожен конкретний користувач зазвичай має справу лише з невеликою частиною всієї бази даних, причому в опрацюванні різними користувачами частини можуть довільним чином перекриватися. Інакше кажучи, кожна БД сприймається її різними користувачами по-різному. Фактично, навіть ті два користувача бази даних, які працюють з одними і тими ж її частинами, можуть мати значно різні уявлення про них.

База даних характеризується моделлю даних, тобто формою організації даних в ній. За типом моделі даних БД діляться на мережеві, ієрархічні і реляційні. В даний час практично застосовується тільки реляційна структура, в якій база даних складається з однієї або декількох двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має такі властивості:

* кожен елемент таблиці - один елемент даних;
* всі осередки в стовпці таблиці однорідні, тобто всі елементи в стовпці мають однаковий тип (числовий, символьний та інше);
* кожен стовпець має унікальне ім'я;
* однакові рядки в таблиці відсутні;
* порядок проходження рядків і стовпців може бути довільним.

Така модель зберігання даних забезпечує зручність використання бази даних на ЕОМ. З огляду на те, що таблиці бази даних можуть бути пов'язані певними відносинами, така модель забезпечує цілісність даних і відсутність надмірності зберігання. Тому вона і використовується в більшості сучасних БД.

Система управління базами даних (СУБД) - спеціалізована програма або комплекс програм, призначена для організації і ведення бази даних. Вона забезпечує зберігання даних і взаємодію користувача з БД, дозволяючи користувачам здійснювати пошук, сортування та вибірку інформації в базі даних, а деяким користувачам - додавати, видаляти і змінювати записи в БД.

Залежно від того, з якою базою даних працює СУБД, вона може бути:

* ієрархічною;
* мережевою;
* реляційною;
* об'єктно-реляційною;
* об'єктно-орієнтованою.

В даний час реляційні СУБД є важливим інструментом в багатьох областях, починаючи з традиційних: бізнес, наукове дослідження, освіта і закінчуючи розробкою пошукових серверів в Internet [6].

1. ВИБІР ТЕХНОЛОГІЇ СТВОРЕННЯ ВЕБ-САЙТІВ

Одним з ключових моментів в розвитку всесвітньої павутини грає веб-розробка — процес створення веб-сайту або веб-додатку. Термін включає розробку додатків електронної комерції, веб-дизайн, програмування для веб на стороні клієнта і серверу, а також конфігурування веб-серверу. Основними етапами веб-розробки є:

* проектування сайту або веб-додатку;
* створення макетів сторінок;
* наповнення;
* обслуговування працюючого сайту або його програмної основи.

HTML (англ. HyperText Markup Language — мова розмітки гіпертекстових документів) —стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

У більшості випадків автор документа суворо визначає зовнішній вигляд документа. HTML дозволяє відзначити, де в документі повинен бути заголовок або абзац за допомогою тегу HTML, а потім документ передається браузеру, що інтерпретує ці теги. Але не всі браузери підтримують деякі теги. Наприклад, один браузер може розпізнавати тег початку абзацу і представляти документ у потрібному вигляді, а інший не має такої можливості і надає документ в один рядок. Користувачі деяких браузерів мають, також, можливість налаштовувати розмір і вид шрифту, колір та інші параметри, що впливають на відображення документа.

HTML - теги можуть бути умовно розділені на дві категорії:

* теги, що визначають, як буде відображатися браузером тіло документа в цілому;
* теги, що описують загальні властивості документа, такі як заголовок чи автор документа.

HTML-документи можуть бути створені за допомогою будь-якого текстового редактора або спеціалізованих HTML-редакторів і конвертерів. Вибір редактора, який буде використовуватися для створення HTML-документів, залежить виключно від поняття зручності і особистих пристрастей кожного автора.

Наприклад, HTML редактори, такі, як «Netscape Navigator Gold» компанії Netscape, дозволяють створювати документи графічно з використанням технології WYSIWYG (What You See Is What You Get). З іншого боку, більшість традиційних засобів для створення документів мають конвертери, що дозволяють перетворювати документи до формату HTML.

Основна перевага HTML полягає в тому, що ваш документ може бути переглянутий в браузерах різних типів і на різних платформах.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) - спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися і до інших видів документів.

Таблицю стилів CSS можна вмонтувати прямо в HTML-сторінку – це внутрішня таблиця стилів. Або ж її можна створити в окремому файлі, і вже потім приєднати посилання на нього до потрібної HTML-сторінки – це зовнішня таблиця стилів. Зовнішню таблицю необхідно підключити до основного HTML- документу за допомогою спеціальних тегів з зазначенням ім'я файлу, що містить таблицю CSS. Завдяки цьому, стиль, описаний у зовнішній таблиці CSS, можна використовувати повторно скільки завгодно разів [11].

З попередньої інформації можна зробити висновки, що використання HTML та CSS є зручним та не складним інструментом, але має ряд недоліків, таких як:

* обмежені можливості при створені сайту;
* відсутність динамічних елементів;
* велика кількість коду.

Оскільки нам потрібний простий та зрозумілий для користувача веб-інтерфейс, приведені недоліки можна проігнорувати.

1. ОГЛЯД ТЕХНОЛОГІЇ JSP

Для того щоб додати динамічності елементам веб-інтерфейсу використаємо технологію Java Server Pages.

JSP представляє технологію, яка дозволяє створювати динамічні веб-сторінки. Спочатку JSP на зорі розвитку Java EE були домінуючим підходом до веб-розробки на мові Java. І хоча в даний час вони поступилися своїм місцем іншій технології – JSF (Java Server Faces), проте JSP продовжують широко використовуватися [12].

По суті Java Server Pages або JSP є HTML-код з вкрапленнями коду Java. У той же час сторінки JSP - це не стандартні HTML-сторінки. Коли приходить запит до певної сторінки JSP, то сервер обробляє її, генерує з неї код HTML і відправляє його клієнту. В результаті користувач після звернення до сторінки JSP бачить в своєму браузері звичайну HTML-сторінку.

Як і звичайні статичні веб-сторінки, файли JSP необхідно розміщувати на веб-сервері, до якого звичайні користувачі можуть звертатися по протоколу HTTP, наприклад, набираючи в адресному рядку браузера потрібну адресу. Однак щоб сервер міг обробляти файли JSP, він повинен використовувати ядро JSP (JSP engine), яке також називають JSP-контейнером. Є безліч движків JSP, і всі вони реалізують одну й ту ж специфікацію і в цілому працюють однаково. Однак тим не менше при перенесенні коду з одного веб-сервера на інший можуть знадобитися невеликі зміни.

В даному випадку для роботи з JSP ми будемо використовувати Apache Tomcat, який одночасно є і веб-сервером і контейнером сервлетів та JSP.

Технологія Java Server Pages містить чотири ключові компоненти:

* директиви (directive) – представляють собою повідомлення для контейнера JSP, що дає можливість визначити параметри сторінки, підключення інших ресурсів, використовувати власні нестандартні бібліотеки тегів;
* дії (actions) – інкапсулюють функціональні можливості в зумовлених тегах, які можна вбудовувати в JSP-сторінки. Дії часто виконуються на основі інформації, що посилається на сервер в складі запиту від певного клієнта. Дії також можуть створювати об'єкти Java для використання їх в скріптлетах JSP;
* скріптлети (scriptlets) – дозволяють вставляти код Java в сторінки JSP, який взаємодіє з об'єктами сторінки при обробці запиту;
* бібліотеки тегів (tag library) – є складовою частиною механізму розширення тегів, що допускає розробку і використання власних тегів [13].

В JSP-сторінку можна вставити досить складну логіку, зробити виклики до бази даних і багато-багато іншого, що необхідно для додатка. Але ось здійснювати вивід на екран всередині самого сервлету - дуже незручно. Саме тому доцільно розділити завдання на дві складові: одна частина обробляє запит, змінює дані, збирає дані, укладає це в якийсь пакет і передає другій частині, яка робить тільки одне - показує ці дані. Для цього ми використовуємо паттерн проектування Model-View-Controller (MVC). Для випадку мого веб-додатку контролером виступають Data Access Object (DAO) класи та сервлети, класи, які описують структуру бази даних - моделлю. А ось на роль уявлення (View) прекрасно підходить JSP - Java Server Pages.

1. ОГЛЯД ТЕХНОЛОГІЇ DAO

З'єднання з базою даних є однією з найважливіших складових програми. Завжди виділяється частина коду, тобто модуль, що відповідає за передачу запитів в БД і обробку отриманих від неї відповідей. У загальному випадку, визначення описує DAO як прошарок між БД і системою. DAO абстрагує сутності системи і робить їх відображення на БД, визначає загальні методи використання з'єднання, його отримання, закриття та (або) повернення в Connection Pool.

Вершиною ієрархії DAO є абстрактний клас або інтерфейс з описом загальних методів, які будуть використовуватися при взаємодії з базою даних. Як правило, це методи пошуку, видалення по ключу, оновлення і т. д.

Реалізація DAO на рівні класу має на увазі використання одного єдиного з’єднання з базою для виклику більш ніж одного методу успадкованого DAO класу.

Певні можливості надаються незалежно від того, який механізм зберігання використовується і без необхідності спеціальним чином відповідати цьому механізму зберігання. Цей шаблон проектування застосовується до безлічі мов програмування, більшості програмного забезпечення, що потребує зберігання інформації і до більшої частини баз даних. Для з’єднання з базою даних використовується певний драйвер, який надається розробником відповідної СУБД [14].

ВИСНОВОК

В результаті проходження переддипломної практики отримано відповіді по темі індивідуального завдання, проаналізовано поняття веб-сервіс хмарного зберігання файлів, проаналізовано мови програмування та обрано основною мову програмування Java, проведено опис технології JavaFX шаблону MVC, було проведено аналіз технологій проектування баз даних та розглянуто шаблон Data Access Object.

Також було поглиблено та закріплено теоретичні та практичні знання, використання яких необхідне при вирішення передбачених програмою практики задач.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

* 1. Что такое облачное хранилище? Доступ до матеріалу: <https://aws.amazon.com/ru/what-is-cloud-storage/>
  2. Онлайновые хранилища данных // ComputerBild : журнал. — 2010. — № 4. — С. 62—67.
  3. Язык программирования Java. Доступ до матеріалу: <https://web-creator.ru/articles/java/>
  4. The Java Tutorials // Oracle Доступ до матеріалу: <https://docs.oracle.com/javase/tutorial/getStarted/intro/changemylife.html>
  5. Плюсы и минусы программирования на Java. Доступ до матеріалу: <http://bit.do/eSifD/>
  6. Дейт. К. Дж. Введення в системи баз даних - Introduction to Database Systems. - 8-е изд. - М.: «Вільямс», 2006.
  7. Драч В.Е., Родионов А.В., Чухраева А.И. Выбор системы управления базами данных для информационной системы промышленного предприятия // Электромагнитные волны и электронные системы. 2018. Т. 23. № 3. С. 71-80.
  8. MySQL. Доступ до матеріалу: <https://uk.wikipedia.org/wiki/MySQL>
  9. Самые популярные СУБД: рейтинг 2018-го года. Доступ до матеріалу: <https://itsource.com.ua/blog/samye-populjarnye-subd-rejting-2018-go-goda/>
  10. HeidiSQL — MySQL, MSSQL and PostgreSQL made easy. Доступ до матеріалу: <https://www.heidisql.com/>
  11. Петлюшкин А.В., HTML в Web-дизайне. – СПб.: БВХ-Петербург, 2004. – 400с.: ил.
  12. JavaServer Pages Доступ до матеріалу: <https://ru.wikipedia.org/wiki/JavaServer_Pages>
  13. Java Server Pages. Что такое JSP. Первое приложение. Доступ до матеріалу: <https://metanit.com/java/javaee/3.1.php>
  14. Data Access Object (DAO). Уровень класса. Доступ до матеріалу: <https://habr.com/ru/post/262243/>