Міністерство освіти і науки України
Національний університет водного господарства та природокористування

Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки
Кафедра обчислювальної техніки

04-04-262М

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з навчальної дисципліни
«**Логічне і функціональне програмування**»
для здобувачів вищої освіти другого (магістерського) рівня
за освітньо-професійною програмою «Комп'ютерна інженерія»
спеціальності 123 «Комп'ютерна інженерія»
денної та заочної форми навчання

Рекомендовано
науково-методичною радою
з якості ННІАКОТ
Протокол № 1 від 09.10.2023 р.

Рівне – 2023

Методичні вказівки до лабораторних робіт з навчальної дисципліни «Логічне і функціональне програмування» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» денної та заочної форми навчання. [Електронне видання] / Бойчура М. В. – Рівне : НУВГП, 2023. – 28 с.

Укладачі: Бойчура М. В., к.т.н., старший викладач кафедри обчислювальної техніки.

Відповідальний за випуск: Круліковський Б. Б., к.т.н., доцент, завідувач кафедри обчислювальної техніки.

Керівник групи забезпечення
спеціальності
123 «Комп'ютерна інженерія» Круліковський Б. Б.

ЗМІСТ

Вступ

Методичні вказівки дисципліни «Логічне і функціональне програмування» є невід'ємною частиною нормативно-методичного забезпечення навчального процесу зі сфери знань 12 «Інформаційні технології». Курс розроблено відповідно до стандарту підготовки магістра з предметної області 123 «Комп'ютерна інженерія». Знання та вміння з дисципліни «Логічне і функціональне програмування» допомагають набути наступні фахові навички: вміння використовувати методи і засоби логічного та функціонального програмування для розв'язання складних і неформалізованих задач, що зустрічаються на практиці; здатність використовувати сучасні методи та мови програмування для розробки алгоритмів і веб-застосунків з урахуванням цілей, обмежень, технічних та економічних аспектів; володіння базовими елементами, що допомагають успішно писати та захищати кваліфікаційну магістерську роботу.

Дисципліна «Логічне і функціональне програмування» вивчається у 2-му семестрі і є ключовою для розв'язання складних і неформалізованих задач, що зустрічаються на практиці; сприяє оволодінню знань та навичок, які необхідні для професійної діяльності з планування, проектування та розробки веб-додатків. Курс передбачає комплексну підготовку фахівця з логічного і функціонального програмування у низці аспектах набуття професійних навичок, визначених у курсі магістра комп'ютерної інженерії.

Метою вивчення дисципліни є набуття студентами знань, необхідних для формування теоретичної бази та практичних навичок використання методів і засобів логічного та функціонального програмування для

розв’язання складних і неформалізованих задач, що зустрічаються в реальних економічних, організаційних і виробничих системах.

Основними цілями викладання даної дисципліни є:

– здобуття навичок розробки застосунків із використанням логіки предикатів;

– навчитись розробляти та розгортати веб-застосунки, що включають роботу з базою даних, авторизацію та автентифікацію із застосуванням саме функціональної парадигми програмування.

В результаті виконання усіх лабораторних робіт студенти повинні:

Знати:

– концепції логічного, функціонального, імперативного та декларативного програмування;

– фази розробки програмного забезпечення;

– особливості мов програмування Prolog та Go;

– переваги використання загалом фреймворків для полегшення розробки програмного забезпечення;

Вміти:

– формалізувати задачі різного рівня складності;

– реалізовувати програми із використанням логіки предикатів Хорна;

– реалізовувати динамічні веб-сайти на мові Go;

– реалізовувати реєстрацію, авторизацію та автентифікацію;

– правильно організувати процес розробки програмного забезпечення;

– проектувати власні додатки;

– програмувати відповідно до принципів Coding Conventions.

Лабораторна робота №1
Програма «Чотирикутник». Програма «Конверт»

Мета

Ознайомитись із основами синтаксису мови Prolog.

Необхідні знання та навички

- елементи логіки предикатів;
- робота в середовищі розробки Visual Prolog v. 5.2;
- основи синтаксису мови Prolog;
- використання тверджень-фактів;
- робота з твердженнями-правилами;
- використання логічних *і*, *або*, *не*, *зворотна імплікація*;
- Coding Conventions;
- структурування програми на мові Prolog;
- знання розділів мови Prolog;
- стандартні доменні типи.

Рекомендований перелік завдань

1. (2 бали) На площині задана деяка множина точок, що позначаються символами ***a, b, c, d,….*** Між деякими із них задані напрямлені зв'язки як відношення виду ***стрілка(точка, точка)***. Причому між окремими точками можуть бути зв'язки в двох напрямках. Відношення ***чотирикутник(точка, точка, точка, точка)*** пов'язує чотири точки, для яких можливий циклічний обхід по наявних стрілках. Побудувати відповідне правило, яке визначає орієнтовані чотирикутники за чотирма стрілками. В програмі задати деяку кількість фактів про наявні стрілки між точками. Знайти всі орієнтовані чотирикутники для заданої множини точок та стрілок.
2. (3 бали) На площині задана множина точок ***a, b, c, d, e***, які сполучені лініями у вигляді відкритого поштового

конверту (рис. 1). Всі лінії сполучення визначені відношенням виду ***лінія(точка, точка)***. Побудувати правило, яке за відомими сполученнями між точками встановлює всі можливі маршрути повного обходу «конверта» таким чином, щоб кожна лінія використовувалася лише один раз. Підказка: правильна кількість маршрутів є більшою, ніж 5.

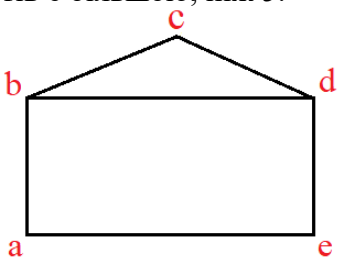

Рис. 1. Схематичне зображення відкритого конверту

Вимоги та рекомендації до програмної реалізації

Для отримання максимальної оцінки розроблені програми повинні враховувати всеможливі випадки вхідних даних в межах постановки задачі. Для оцінки добре – допускаються невеликі неточності у побудованих правилах. Для задовільної оцінки необхідно врахувати хоча б по 2 види умов для кожної з задач.

Контрольні запитання

1. Якими двома способами позначається логічне *і* в Prolog?
2. Якими двома способами позначається логічне *або* в Prolog?
3. Якими двома способами позначається *зворотна імплікація* в Prolog?
4. Чим відрізняється оголошення твердження-факту від твердження-правила?

5. Яких основних принципів Coding Convention варто дотримуватись при програмуванні на Prolog?

6. Яка структура програми мови Prolog?

7. Які стандартні доменні типи Ви знаєте?

8. Чи еквівалентними є твердження-факти: line(a, b) та line(b, a)?

9. Яким чином можна заборонити повторне проходження по деякій лінії у наведених вище задачах?

10. Чим відрізняється пошук орієнтованих трикутників від пошуку орієнтованих чотирикутників?

Лабораторна робота №2
Складені правила. Домени з альтернативами

Мета

Навчитись формулювати складені правила та використовувати домени з альтернативами.

Необхідні знання та навички

- складені правила;
- твердження-відсікання;
- алгоритмічні конструкції;
- механізм відкату;
- точки розгалуження;
- уніфікаційні підпрограми Prolog;
- змінні у Prolog;
- вільна змінна;
- змінна зі значенням;
- складені домени;
- випадки застосування складених доменів;
- оголошення домена з альтернативами;
- використання доменів з альтернативами.

Рекомендований перелік завдань

1. Оголосити відношення виду *власник(ім'я, річ)*, де другий об'єкт *річ* є функтором складеного домену із альтернативами: *транспорт(вид, модель, рік_випуску)* або *нерухомість(вид, адреса, площа)* або *колекція(предмет, кількість, вартість)*. Тут об'єкти *ім'я, вид, модель, адреса, предмет* належать домену рядків символів string; об'єкти *рік_випуску, кількість* є цілими числами домену integer; об'єкти *площа, вартість* є дійсними числами домену real. Для зручності виконання складних запитів рекомендується оголосити також відношення *Osoba(string)*. В програмі

задати деяку кількість фактів про власників транспортних засобів, нерухомості та колекцій.

Продемонструвати виконання програми на наступних прикладах:

- (1 бал) Пошуку осіб, хто підтримує (колекціонує) страусів;
- (1,5 бали) Пошуку всіх власників понад 3-х автомобілів;
- (2,5 балів) Пошуку всіх власників ретро-автомобілів, що проживають у власному будинку у місті Рівне і колекціонують метеликів.

Вимоги та рекомендації до програмної реалізації

Для отримання максимальної оцінки розроблена програма повинна враховувати всеможливі випадки вхідних даних в межах постановки задачі. Для оцінки добре – допускаються невеликі неточності у побудованих правилах. Для задовільної оцінки необхідно, щоб розв'язки були хоча б на 50% вірними при різних варіантах вхідних даних.

Контрольні запитання

1. Чим відрізняються складені правила від нескладених правил?
2. У яких випадках доцільно користуватись твердженнями-відсіканнями?
3. Як працює механізм відкату?
4. Як пов'язані точки розгалуження із механізмом відкату?
5. Які Ви знаєте обмеження при використанні змінних у Prolog?

6. Яким чином можна змінити стан вільної змінної, щоб вона стала змінною зі значенням? Чи можлива зміна стану у зворотній бік?

7. Скільки альтернатив може мати домен з альтернативами?

8. Яка користь від предикатів виду *Osoba(string)* при формуванні цільових запитів в межах даної лабораторної роботи?

9. Яким чином доцільно користуватись анонімними змінними?

10. Поясніть принцип перебору всеможливих тверджень (фактів і правил), що реалізують уніфікаційні підпрограми Prolog?

Лабораторна робота №3-4
Повторювані обчислення

Мета

Навчитись використовувати основні підходи до реалізації повторюваних обчислень на мові Prolog.

Необхідні знання та навички

- способи управління повторюваними обчисленнями;
- механізм відкату;
- способи управління відкатом;
- взаємодія відкату після невдачі та відкату із відсіканням;
- реалізація меню в Prolog;
- відлагодження програми на Prolog;
- рекурсія;
- випадки, які приводять до використання рекурсії;
- повторення на прямому ході рекурсії;
- повторення на зворотному ході рекурсії;
- обчислення факторіалу, суми з умовами, суми ряду.

Рекомендований перелік завдань

1. (2 бали) Знайти суму $\sum_{i=a}^{b} i + \sum_{i=a}^{10b} i^2$. Тут $a, b \in \mathbb{Z}$.
2. (2 бали) Знайти значення $n!$ Тут $n \in \mathbb{Z}_0$.
3. (2 бали) Обчислити n-й член послідовності Фібоначчі. Тут $n \in \mathbb{Z}$.
4. (4 бали) Знайти із точністю 10^{-4} розклади функцій $e^x \approx \sum_{i=0}^{n} \frac{x^i}{i!}$, $\sin x \approx \sum_{i=0}^{n} (-1)^i \frac{x^{2i+1}}{(2i+1)!}$, $\cos x \approx \sum_{i=0}^{n} (-1)^i \frac{x^{2i}}{(2i)!}$ в ряд Тейлора. Тут $x \in \mathbb{Z}$, $n \in \mathbb{Z}_0$.

Вимоги та рекомендації до програмної реалізації

Для безпосереднього виконання даних завдань застосовувати рекурсію. При представленні прикладів функціонування програми користуватись меню (реалізувати його із застосуванням повторень, що визначені користувачем).

Щоб отримати максимальну оцінку розроблені програми повинні враховувати всеможливі випадки вхідних даних в межах постановок задач. Для оцінок добре та задовільно – допускаються невеликі неточності у побудованих правилах або правильне виконання не всіх задач.

Контрольні запитання

1. Які Ви знаєте способи управління повторюваними обчисленнями та яка між ними різниця?
2. Які Ви знаєте способи управління відкатом?
3. Як реалізувати меню в Prolog?
4. Яка відмінність розміщення фактів у ПВП і рекурсії?
5. Яка принципова відмінність повторень на прямому та зворотному ходах рекурсії?
6. Яка умова завершення рекурсії при обчисленні факторіалу?
7. Яка умова завершення рекурсії при обчисленні суми з умовами?
8. Яка умова завершення рекурсії при обчисленні суми ряду?
9. Наскільки важко переробити рекурсивний розрахунок синуса при наявних твердженнях для обчислення косинуса?
10. Що потрібно змінити у твердженнях, щоб перетворити ПВП на рекурсію?

Лабораторна робота №5
Елементарні дії над елементами списку

Мета

Навчитись організовувати основні елементарні дії над елементами списку.

Необхідні знання та навички

- синтаксис роботи зі списками;
- оголошення списку;
- цільові запити до списків;
- розділення списку та голову та хвіст;
- основні алгоритми роботи зі списками;
- наповнення, вивід, пошук, вставка та видалення елементів.

Рекомендований перелік завдань

1. Сформувати список елементів деякого типу (цілі числа) та виконати із ним наступні дії:

a) (1 бал) перегляд списку з виведенням на екран;
b) (1 бал) пошук першого входження елемента із заданим значенням;
c) (1 бал) вставка нового елемента в список на початок, в кінець, перед деяким значенням, після деякого значення;
d) (1 бал) вилучення елемента із списку;
e) (1 бал) поділ списку на два за певною ознакою.

Вимоги та рекомендації до програмної реалізації

Щоб отримати максимальну оцінку розроблені програми повинні враховувати всеможливі випадки вхідних даних в межах постановок задач. Для оцінок добре та

задовільно – допускаються невеликі неточності у побудованих правилах або правильне виконання не всіх задач.

Контрольні запитання

1. Що таке список?
2. Як оголосити список?
3. За яким принципом ініціалізуються вкладені списки?
4. Як вивести вкладений список?
5. Як працює операція РСГХ?
6. Чи можливо відділити одночасно 3 голови від списку?
7. Які з основних алгоритмів роботи зі списками є складнішими/простішими?
8. Яким чином доцільно використовувати твердження-відсікання при реалізації основних алгоритмів роботи зі списками?
9. Які можливі випадки виконання даної операції Res = (H|T) ?
10. Що має відбуватись у точці зупинки рекурсії при виконання операції вставки елемента у певну позицію списку?

Лабораторна робота №6
Сортування елементів списку

Мета

Набути навички сортування елементів списку.

Необхідні знання та навички

- вступ;
- стандартні твердження для роботи з рядками;
- твердження str_len, concat, frontchar, frontstr, str-int, str-real, str-char, upper-lower, isname, fronttoken;
- основні алгоритми для роботи з рядками;
- пошук, вставка, видалення, конвертація символів/рядків.

Рекомендований перелік завдань

1. Сформувати список елементів деякого типу (цілі числа) та впорядкувати його елементи за такими правилами: включення; вибір; обмін.

Вимоги та рекомендації до програмної реалізації

Щоб отримати максимальну оцінку розроблені програми повинні враховувати всеможливі випадки вхідних даних в межах постановок задач. Для оцінок добре та задовільно – допускаються невеликі неточності у побудованих правилах або правильне виконання не всіх задач.

Контрольні запитання

1. Які відмінності між обробкою елементів списків та рядків?
2. Які Ви знаєте стандартні алгоритми роботи з рядками?

3. Для яким трьох цілей можна використати твердження frontchar ?

4. Як конвертувати рядок в число?

5. Що таке атом?

6. Які Ви знаєте основні алгоритми роботи з рядками?

7. Як вставити символ в кінець списку?

8. Якими двома способами можна перетворити усі букви рядка у верхній регістр?

9. Яким чином можна запобігти спробі видалення символа з неіснуючої позиції рядка?

10. Як перетворити (зашифрувати) вихідний текст у новий, в якому кожен символ замінений наступним із таблиці ASCII?

Лабораторна робота №7-8
Розробка веб-сайту з використанням фреймворку Bootstrap та мови Go

Мета

- набути базові практичні навички використання фреймворку Bootstrap;
- навчитись основам розробки веб-сайтів на мові Go;
- викласти веб-сайт на хостинг.

Необхідні знання та навички

- пакети net/http, html/template;
- структура веб-додатку;
- запуск локального сервера;
- система маршрутизації;
- передача даних на веб-сторінку;
- Bootstrap;
- файлова структура веб-додатку;
- доступ до статичних файлів CSS та JavaScript;
- створення шаблонів;
- використання шаблонів;
- шаблонізатори;
- http.Handler;
- передача масивів на html-сторінку;
- умовні конструкції в шаблонізаторах;
- оператори в шаблонізаторах.

Рекомендований перелік завдань

1. (1 бал) Підняти сервер і вивести будь-яку веб-сторінку.

2. (4 бали) У відповідності до номера варіанту знайти готову веб-сторінку, сформовану за допомогою фреймворку Bootstrap, та інтегрувати її в проект.

3. (1 бал) Використати шаблонізатори.

4. (1 бал) Наповнити даними сумарно 2 вкладки із використанням Bootstrap.

5. (3 бали) Вимістити розроблений веб-сайт на будь-якому хостингу.

Варіанти до виконання поставленого завдання

Варіант 1. Потрібно створити сторінку із автобіографією. Там описати свої хобі (із фотографіями), результати поточної успішності (у вигляді таблиці) і т.д.

Варіант 2. Розробити сторінку інтернет-магазину. Тут описати інформацію про наявні продукти (із фотографіями), способи доставки, вартість товарів (у вигляді таблиці) і т.д.

Варіант 3. Розробити сторінку кафедри. Тут описати інформацію про викладачів (із фотографіями), предмети, які викладають викладачі кафедри (у вигляді таблиці) і т.д.

Варіант 4. Розробити сторінку аптеки. Тут описати перелік ліків (із фотографіями), їх кількості та ціни (у вигляді таблиці) і т.д.

Варіант 5. Розробити сторінку футбольного комітету. Тут описати керівний склад комітету (із фотографіями), список команд з додатковою інформацією (у вигляді таблиці) і т.д.

Варіант 6. Розробити сторінку громадської організації. Тут описати основних учасників організації (із фотографіями), напрямки діяльності з вказанням відповідальних учасників (у вигляді таблиці) і т.д.

Варіант 7. Розробити сторінку студентського гуртка. Тут описати основних учасників гуртка (із фотографіями), розклад заходів у поточному році (у вигляді таблиці) і т.д.

Варіант 8. Розробити сторінку косметичної компанії. Тут описати наявність розповсюджуваної продукції (із фотографіями), опис характеристик та ціну (у вигляді таблиці) і т.д.

Варіант 9. Розробити сторінку політичної організації. Тут описати керівний склад організації (із фотографіями), напрямки діяльності та відповідальних виконавців (у вигляді таблиці) і т.д.

Варіант 10. Розробити сторінку з заголовком «Олімп». Тут описати грецьких або римських богів (із фотографіями), сфери відповідальності кожного з них (у вигляді таблиці) і т.д.

Варіант 11. Розробити сторінку інформаційного відділу підприємства. Тут описати персональний склад відділу (із фотографіями), перелік послуг з вказанням їх вартості (у вигляді таблиці) і т.д.

Варіант 12. Розробити сторінку Вашої студентської групи. Тут описати перелік студентів, які навчаються у даній академічній групі (із фотографіями), персональні дані (номер залікової крижки, дата народження, адреса проживання – у вигляді таблиці) і т.д.

Варіант 13. Розробити сторінку банку. Тут описати базу даних клієнтів банку (із фотографіями), наявні послуги з вказанням цінової політики (у вигляді таблиці) і т.д.

Варіант 14. Розробити веб-сторінку кінотеатру. Тут описати перелік фільмів у прокаті (із фотографіями), прайс-лист (у вигляді таблиці) і т.д.

Варіант 15. Розробити сторінку асоціації кінолюбителів. Навести перелік членів асоціації (із фотографіями), перелічити переможців різних кінопремій за останні 5 років (у вигляді таблиць), розмістити на сторінці анонси найближчих відомих фільмів.

Варіант 16. Розробити сторінку «Сайт наукових новин». Тут описати поточні новини (із фотографіями), склад команди журналістів з інформацією про них (у вигляді таблиці) і т.д.

Варіант 17. Розробити сторінку агентства нерухомості. Розмістити на ній всі наявні пропозиції щодо продажу та купівлі житла (із фотографіями), навести перелік ріелторів, їх номери телефонів та досвід роботи.

Варіант 18. Розробити сторінку автомобільного салону. Навести перелік автомобілів з їх фотографіями та цінами (у вигляді таблиці), розмістити інформацію із подяками від задоволених покупців тощо.

Варіант 19. Розробити сторінку асоціації книголюбів. Тут навести перелік її осередків у країні, склад кожного з осередків (із фотографіями), перелічити рекомендовані асоціацією книги для прочитання (у вигляді таблиці) тощо.

Варіант 20. Розробити сторінку студентської ради. Тут навести склад ради і перелік асоційованих членів із фотографіями (у вигляді таблиці), джерела фінансування, проекти ухвалених рішень, статут тощо.

Вимоги та рекомендації до програмної реалізації

Інформація на веб-сторінці має виглядати зв’язно. Сторінка має обов’язково складатись хоча б з трьох частин: header, content, footer. Для отримання максимальної оцінки студенти повинні виконати увесь рекомендований перелік завдань. Для оцінок добре та задовільно – допускаються незначні відхилення від постановок задач та виконання не всіх завдань.

Для отримання мінімальної оцінки потрібно пояснити усі рядки коду, наведеного в файлі *.go.

Контрольні запитання

1. Чим корисним є фреймворк Bootstrap?
2. Яка типова процедура викладення сайту на хостинг?
3. Як підняти сервер на мові Go?
4. Як передати дані з файлу *.go на веб-сторінку?
5. Що таке шаблонізатори?
6. Чи можна якимось чином використовувати оператори вибору у коді файла *.html
7. Як найбільш правильно створювати нові веб-сторінки деякого сайту, не рушачи загальну стилізацію сайту?
8. Чи можна якимось чином CSS чи JavaScript у проекті, написаному на мові Go?
9. Що таке html/template ?
10. Які зазвичай є обмеження та безкоштовних хостингах?

Лабораторні роботи №9-10
Інтеграція баз даних у веб-додаток

Мета

Навчитись реалізовувати CRUD-операції сумісно засобами мови Go, фреймворку Bootstrap та баз даних.

Необхідні знання та навички

- створення бази даних;
- підключення до бази даних;
- додавання, зчитування, редагування та видалення даних з бази даних;
- обмін даними між базою даних та html-сторінкою;
- об’єкт типу *http.Request;
- відправлення форм;
- методи POST та GET;
- роутер Gin.

Рекомендований перелік завдань

1. Потрібно інтегрувати у веб-сайт, розроблений на лабораторній роботі №7-8, можливість доступу до бази даних. А саме:

a) (2 бали) реалізувати додавання даних;
b) (2 бали) реалізувати зчитування даних;
c) (3 бали) реалізувати редагування даних;
d) (3 бали) реалізувати видалення даних.

Вимоги та рекомендації до програмної реалізації

Інформація на веб-сайті має виглядати зв’язно. Для отримання максимальної оцінки студенти повинні виконати увесь рекомендований перелік завдань. Для оцінок добре та задовільно – допускаються незначні відхилення від постановок задач та виконання не всіх завдань.

Для отримання мінімальної оцінки потрібно наступне: викласти сайт на хостинг, а також пояснити усі рядки коду, наведеного в файлах *.go та *.html.

Контрольні запитання

1. Які Ви знаєте CRUD-операції?
2. Чим відрізняються методи GET та POST?
3. Для чого призначений роутер Gin?
4. Які серед CRUD-операцій є простішими/складнішими у реалізації?
5. Як підключити базу даних до веб-проекту на мові Go?
6. Як правильно користуватись формами в мові Go?
7. Що таке *http.Request ?
8. Як реалізується обмін даними між базою даних та html-сторінкою?
9. Як видалити певні дані за сторінки із використанням CRUD-операцій?
10. Яким чином можна передати дані з однієї сторінки на іншу?

Лабораторні роботи №11-12
Реалізація авторизації та автентифікації користувачів

Мета

Навчитись реалізовувати реєстрацію, авторизацію та автентифікацію. У результаті має бути викладений на хостинг повноцінний веб-сайт, інтерфейс якого дозволяє задовольняти типові потреби як адміністратора, так і користувачів.

Необхідні знання та навички

- маршрутизація gorilla/mux;
- рядок запиту;
- REST API;
- структура JWT;
- header;
- playload;
- signature;
- авторизація;
- автентифікація;
- ролі.

Рекомендований перелік завдань

1. Доповнити веб-сайт, розроблений на попередніх лабораторних роботах, наступними можливостями:

a) (3 бали) реєстрація користувача;
b) (3 бали) автентифікація користувача;
c) (4 бали) робота з ролями користувачів.

Вимоги та рекомендації до програмної реалізації

Інформація на веб-сайті має виглядати зв’язно. Для отримання максимальної оцінки студенти повинні виконати увесь рекомендований перелік завдань. Для оцінок добре та

задовільно – допускаються збої у роботі сайту та виконання не всіх завдань.

Для отримання мінімальної оцінки потрібно наступне: викласти сайт на хостинг, а також пояснити усі рядки коду, наведеного в файлах *.go та *.html.

Контрольні запитання

1. Що таке REST API?
2. Яка різниця між авторизацією та автентифікацією?
3. Що таке JWT?
4. В чому перевага організації роботи з ролями?
5. Що таке gorilla/mux?
6. Що має бути в інтерфейсі адміністратора?
7. Чого не має бути в інтерфейсі звичайного авторизованого користувача?
8. Що таке Cookie?
9. Яка оптимальна кількість таблиць бази даних при роботі з ролями користувачів?
10. Як розділити доступ до інструментів сайту для різних ролей?

Рекомендована література

Основна

1. Заяць В. М., Заяць М. М. Логічне і функціональне програмування. Системний підхід : підручник /2-ге видання, випр. та доповн. Рівне : Національний університет водного господарства та природокористування, 2018. 422 с.
2. Дейнега Л. Ю., Камінська Ж. К., Левада І. В., Сердюк С. М. Практичне програмування мовою Visual Prolog : навч. посіб. Запоріжжя : ЗНТУ, 2016. 236 с.
3. Темнікова О. Л. Математична логіка та теорія алгоритмів: конспект лекцій : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 177 с.
4. Безсмертний І. А. Інтелектуальні системи. URL: https://stud.com.ua/139940/informatika/intelektualni_sistemi (Last access: 14.01.2023).
5. Buscaroli R., Chesani F., Giuliani G., Loreti D., Mello P. A Prolog application for reasoning on maths puzzles with diagrams. Journal of Experimental & Theoretical Artificial Intelligence. 2022. DOI: 10.1080/0952813X.2022.2062456.
6. Зуєв А. О., Сальніков Д. В., Васильченков О. Г. Методичні вказівки для виконання самостійних робіт «Основи програмування на мові Golang». Харків : НТУ «ХПІ», 2022. 35 с.
7. Freeman A. Pro Go The: Complete Guide to Programming Reliable and Efficient Software Using Golang. First Edition. London : Apress Media, 2022. 1076 p.
8. Bates M., LaNou C., Raymond T. How to Code in Go. New York : DigitalOcean, 2020. 447 p.

Додаткова

1. Wampler D. Programming Scala. Thitd Edition. Sebastopol : O'Reilly Media, Inc. 2021. 520 p.

2. A Tour of Go. URL: https://go.dev/tour/welcome/1 (Last accessed: 02.06.2023).

3. Documentation - The Go Programming Language. URL: https://go.dev/doc/ (Last accessed: 02.06.2023).

4. Go by Example. URL: https://gobyexample.com/ (Last accessed: 02.06.2023).

5. Standard library - Go Packages. URL: https://pkg.go.dev/std (Last accessed: 02.06.2023).

6. Introduction | Tour of Scala | Scala Documentation. URL: https://docs.scala-lang.org/tour/tour-of-scala.html (Last accessed: 02.06.2023).

7. A Scala Tutorial for Java Programmers | Scala Documentation. URL: https://docs.scala-lang.org/tutorials/scala-for-java-programmers.html (Last accessed: 02.06.2023).

8. Introduction | Scala 3 — Book | Scala Documentation. URL: https://docs.scala-lang.org/scala3/book/introduction.html (Last accessed: 02.06.2023).

9. What is F# | Microsoft Learn. URL: https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp (Last accessed: 02.06.2023).