

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ЗАТВЕРДЖУЮ

Голова приймальної комісії

В. С. Пономаренко

«*В.С.*» квітня 2019р.



ПРОГРАМА

фахового вступного випробування
освітній ступінь «МАГІСТР»

спеціальність 122 «Комп'ютерні науки»
освітньо-професійна програма «Комп'ютерні науки»

Харків, 2019

Фахове випробування є комплексним іспитом, що включає практичні завдання (три завдання різного ступеню складності) в межах галузевого стандарту вищої освіти України з підготовки бакалаврів галузі знань 12 “Інформаційні технології”, спеціальність 122 “Комп’ютерні науки та інформаційні технології”. До складу завдань внесені питання за наступними нормативними навчальними дисциплінами: об’єктно-орієнтоване програмування, організація баз даних і знань, технології паралельних обчислень.

Завдання фахового випробування складено з метою виявлення знань, вмінь, компетентностей, якими володіє бакалавр за галуззю знань 12 “Інформаційні технології” (табл. 1).

Таблиця 1

Основні компетентності, якими повинен володіти бакалавр за галуззю знань 12 “Інформаційні технології”

Професійні компетентності
<i>Загально-професійні:</i>
- ґрунтовна підготовка в області програмування, володіння алгоритмічним мисленням, методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик
- ґрунтовна математична підготовка, а також підготовка з теоретичних, методичних і алгоритмічних основ інформаційних технологій для використання математичного апарату під час вирішення прикладних і наукових завдань в області інформаційних систем і технологій
- здатність до проектної діяльності в професійній сфері, вміння будувати і використовувати моделі для опису об’єктів і процесів, здійснювати їх якісний аналіз
- володіння технологією розроблення програмного забезпечення відповідно до вимог і обмежень замовника
<i>Спеціалізовано-професійні:</i>
- здатність до математичного та логічного мислення, знання основних понять, ідей і методів фундаментальної математики та вміння їх використовувати під час розв’язання конкретних завдань

Закінчення табл. 1

- знання сучасних методів побудови та аналізу ефективних алгоритмів і вміння їх реалізувати в конкретних застосуваннях
- знання принципів структурного програмування, сучасних процедурно-орієнтованих мов, основних структур даних і вміння їх застосовувати під час програмної реалізації алгоритмів професійних завдань
- здатність до об'єктно-орієнтованого мислення, знання об'єктно-орієнтованих мов програмування та вміння застосовувати об'єктно-орієнтований підхід під час проектуванні складних програмних систем
- знання принципів і правил формалізації економічних ситуацій, вміння застосувати математичні методи обґрунтування та прийняття управлінських і технічних рішень у різних ситуаціях
- знання сучасних технологій та інструментальних засобів розробки програмних систем, вміння їх застосовувати на всіх етапах життєвого циклу
- знання сучасних теорій організації баз даних та знань, методів і технологій їх розроблення за такими компетентностями: <ol style="list-style-type: none"> 1) концепції та принципи організації БнД, БД та БЗ; 2) місце БнД у інформаційних системах (ІС); 3) рівні абстракції даних при проектуванні БД; 4) вимоги до БД і методи їхнього забезпечення; 5) загальні принципи побудови системи управління базами даних та структуру і принципи її функціонування; 6) основні моделі та мовні засоби СКБД; 7) принципи організації структур збереження даних і методи доступу до даних; 8) відмінності між транзакційною та аналітичною БД; 9) етапи проектування БД; 10) методи розробки логічної й фізичної моделей і фізичної схеми даних у середовищі сучасних CASE-інструментів та засобами СКБД; 11) принципи взаємодії прикладних програм, які виконані на мові високого рівня, з реляційними системами управління базами даних; – вміння проектувати логічні та фізичні моделі баз даних і запити до них, включаючи такі компетентності: <ol style="list-style-type: none"> 1) формулювати вимоги до БД і забезпечувати її властивості; 2) проектувати концептуальну модель даних конкретної предметної

області;

3) вибирати СКБД у процесі технічного проектування на основі оціночних варіантів баз даних, вимог користувачів, аналізу технічних, економічних, функціональних, сервісних характеристик СКБД, використовуючи науково-технічну, довідкову інформацію;

4) розробляти логічну структуру бази даних у процесі технічного проектування за допомогою методу нормалізації відношень, використовуючи методи реляційної алгебри, рівні абстракції даних, вимоги вибраної СКБД;

5) розробляти таблиці баз даних і зв'язок між ними в умовах технічного проектування за допомогою відповідного технічного і програмного забезпечення, використовуючи конструктори таблиць;

6) розробляти фізичну структуру бази даних у процесі робочого проектування за допомогою вибраної СКБД, використовуючи сучасні технічні і програмні засоби розробника баз даних;

7) експортувати дані до зовнішнього середовища і імпортувати дані із зовнішнього середовища у бази даних у процесі інтегрування програмного забезпечення користувача у інформаційні вузли Internet за допомогою сучасного технічного і програмного забезпечення, використовуючи технології розподілених застосувань, WEB-технології та ін.;

8) створювати таблиці баз даних, тригери, збережені процедури, індекси в умовах розробки баз даних за допомогою програмних і технічних засобів проектування баз даних, використовуючи візуальні інструменти інтегрованих оболонок розробника програмного забезпечення;

9) розробляти введення, модифікацію, вилучення, відображення даних у таблиці бази даних у процесі робочого проектування за допомогою технічних і програмних засобів, використовуючи форми введення та модифікації даних і табличні режими;

10) розробляти навігації по набору даних в умовах доступу до потрібного запису набору даних за допомогою програмних засобів СКБД, використовуючи навігаційні методи об'єктів;

розробляти методи сортування, фільтрації, пошуку даних у процесі відбору потрібних даних, що відповідають будь-яким критеріям, за допомогою програмних засобів СКБД, використовуючи методи зміни індексів полів.

знання сучасних класифікацій паралельних обчислювальних систем (класифікації Флінна та Джексона), принципи побудови архітектур паралельних обчислювальних систем за класифікацією Флінна, що визначають роботу з потоками даних та команд (завдань)
знання принципів моделі паралельних обчислювальних процесів, концепцію необмеженого паралелізму
знання принципів функціонування багатопроцесорних систем зі загальною і розподіленою пам'яттю, архітектур систем зі загальною та розподіленою пам'яттю, векторно-конвейєрних систем, систем з масовим паралелізмом та обчислювальних кластерів
знання паралелізму даних і паралелізму завдань, показників ефективності розпаралелювання: прискорення, ефективність паралельного алгоритму;
знання базової моделі алгоритму у вигляді графа "операнд - операції", уявлення алгоритму у вигляді графа потоку даних, розклад паралельних обчислень
знання принципів побудови паралельних методів та алгоритмів: декомпозиція завдання на підзавдання; виокремлення інформаційних залежностей між підзавданнями; масштабування підзавдань; розподіл підзавдань між процесорами системи
знання основи технології та загальної схеми виконання паралельної програми при використанні технології OpenMP – схема Fork/Join
знання структури OpenMP: директиви, бібліотека функцій, набір змінних оточення.
знання областей видимості директив OpenMP: паралельний фрагмент, паралельна область, паралельна секція; формату директив OpenMP
знання методів розпаралелювання в OpenMP: розпаралелювання за циклами та за секціями
знання визначення загальних і локальних змінних, сумісної обробки локальних змінних (операція редукції)
знання засобів синхронізації, методів управління кількістю потоків в OpenMP
знання функцій бібліотеки OpenMP для створення паралельних програм
знання основних принципів роботи MPI (інтерфейсу передачі даних); парних (point-to-point) операцій між двома процесами і колективних (collective) комунікаційних дій для одночасної взаємодії декількох процесів; поняття

комунікаторів
знання основних функцій взаємодії паралельних процесів в MPI
знання структури паралельної програми, операцій ініціалізації та завершення MPI-програми
знання операцій передачі даних від одного процесу всім процесам програми; передачі даних від всіх процесів одному процесу (операції редукції; базових типів операцій MPI для функцій редукції даних)
знання блокуючих точкових функцій передачі повідомлень; функції одночасного виконання передачі і прийому повідомлень в MPI
знання призначення та складу колективних операцій передачі даних в MPI
знання призначення та складу глобальних обчислювальних операцій над розподіленими даними; склад та призначення глобальних операцій редукції в MPI
знання методів управління групами процесів і комунікаторами; операції щодо створення груп процесів та отримання інформації про групи процесів в MPI
знання операцій зі створення та управління комунікаторами в MPI
знання складу та призначення похідних типів даних в MPI

СТРУКТУРА ЕКЗАМЕНАЦІЙНОГО БІЛЕТА З ФАХОВОГО ВИПРОБУВАННЯ

Розподіл балів відповідно до рівня складності завдань у білеті з фахового випробування наведено у табл. 2.

Таблиця 2

Розподіл балів відповідно до рівня складності завдань
у білеті з фахового випробування

Рівень складності	Кількість контрольних завдань		Кількість балів за одне завдання	Розподіл балів в одному білеті	Різновид завдань
	Тестові завдання	задачі			
I. Рівень складності	–	1	1 діагностичне завдання	70	програмування
II. Рівень складності	–	1	1 евристичне завдання	70	проектування бази даних предметної області
III. Рівень складності	–	1	1 евристичне завдання	60	технології паралельних обчислень

Практичні завдання включають завдання різного рівня складності, а саме одне діагностичне завдання та два евристичних завдання, загальна оцінка за виконання яких складає 100 балів.

ЗМІСТ ФАХОВИХ ВСТУПНИХ ВИПРОБУВАНЬ

Тема 1. Основні положення об'єктно-орієнтованого підходу

1.1. Прості та складні програмні системи.

Декомпозиція програмних систем. Способи декомпозиції. Об'єктно-орієнтована декомпозиція. Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом. Об'єктно-орієнтований аналіз та його мета. Поняття предметної області. Головні види вимог до програмної системи.

1.2. Об'єктно-орієнтоване проектування.

Елементи архітектури додатку. Визначення класів предметної області. Принципи проектування класів. Об'єктно-орієнтоване програмування. Прин-

ципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм.

Тема 2. Введення до платформи Microsoft .NET та мови C#

2.1. Платформа Microsoft .NET.

Архітектура платформи Microsoft .NET , засоби розроблення додатків, компіляція та виконання програм, бібліотека базових класів, система типізації.

2.2. Загальні відомості про мову C#:

Особливості використання мови, алфавіт, типи даних, порівняння типів-значень та типів-посилань, вбудовані типи-значення, вбудовані типи-посилання, одновимірні та багатовимірні масиви, операції, оператори, структура програми, коментарі, особливості використання функцій, механізми передачі параметрів, простори імен, основи використання бібліотеки базових класів .NET

Тема 3. Реалізація головних концепцій об'єктно-орієнтованого програмування у мові C#

3.1. Абстрактні типи даних.

Проектування Абстрактного типу даних. Синтаксис структур та класів у мові C#. Елементи класу. Доступ до елементів класу. Посилання this. Перевантаження методів класу.

3.2. Об'єкти в програмі.

Послідовність створення об'єкта. Конструктори. Основні властивості конструкторів. Звільнення пам'яті. Система "збору сміття". Статичні дані та методи: призначення, властивості, особливості використання. Відношення агрегації. Реалізація агрегації у мові C#. Відношення спадкування. Синтаксис спадкування у мові C#. Ініціалізація об'єкта базового класу. Варіанти використання спадкування. Перевизначення методів. Заборона спадкування. Рядкове представлення об'єкта.

3.3. Реалізація принципу поліморфізму у мові C#.

Раннє та пізнє зв'язування. Віртуальні методи. Абстрактні класи та методи. Реалізація поліморфної поведінки на базі абстрактного класу. Правила застосування абстрактних класів. Інтерфейси. Реалізація поліморфної поведінки на базі інтерфейсу. Правила застосування інтерфейсів. Принципи перевантаження операцій. Особливості використання функції operator. Індексатори. Властивості.

3.4. Обробка виняткових ситуацій

Види помилок у програмах. Проблеми традиційного підходу до обробки помилок. Механізм обробки виключень. Класи виключень стандартної бібліотеки .NET. Синтаксис обробки виключень. Перевірка на арифметичне переповнення.

Тема 4. Стандартні бібліотеки класів Microsoft .NET Framework

4.1. Джерела та споживачі даних.

Загальні відомості про потоки введення-виведення даних. Алгоритми роботи потоків введення-виведення даних. Основні класи стандартної бібліотеки .NET для підтримки введення-виведення даних.

4.2. Колекції.

Загальні відомості про колекції. Основні елементи та структури даних стандартної бібліотеки колекцій .NET Типізовані колекції.

4.3. Рядковий тип даних.

Особливості реалізації рядкового типу даних у платформі .Net. Класи стандартної бібліотеки .NET для подання рядків та особливості їхнього використання. Форматування рядків. Призначення та застосування регулярних виразів. Підтримка регулярних виразів у стандартній бібліотеці .NET. Спеціальні символи, які використовуються у регулярних виразах.

4.4. Атрибути.

Введення до атрибутів. Елементи програми до яких можливо застосування атрибутів. Визначені атрибути. Використання атрибутів умовної компіляції. Атрибути рівня модулю компіляції.

4.5. Збереження та відновлення стану об'єктів у .NET

Серіалізація та десеріалізація. "Граф" об'єктів при серіалізації. Створення класів, об'єкти яких можливо серіалізувати. Процеси серіалізації та десеріалізації. Формати серіалізації. Серіалізація та десеріалізація об'єктів у двійковому та XML-форматах.

Тема 5. Розроблення DLL-бібліотек

5.1. Бібліотеки та їх використання. Статичні та динамічні бібліотеки. DLL-бібліотеки.

5.2. Розроблення DLL-бібліотеки на платформі Microsoft .NET.

Тема 6. Основи програмування, керованого подіями

6.1. Делегати та події.

Загальні відомості про делегати. Оголошення та використання делегатів у мові C#. Анонімні методи. Групові делегати. Загальні відомості про події. Генерування подій.

6.2. Основи використання технології Windows Forms

"Традиційна" модель програмування на платформі .NET. Модель "Windows-програмування" на платформі .NET Технологія Windows Forms. Форми. Загальна структура додатку з графічним інтерфейсом користувача на платформі .NET. Розроблення додатків Windows Forms за допомогою інтегрованого середовища. Події рівня форми.

Тема 7. Розроблення графічних інтерфейсів користувача

7.1. Основи архітектури додатків Windows Forms.

Модель подій у Windows Forms. Діалогові вікна. Основні візуальні елементи управління: властивості та використання. "Колекція" візуальних елементів управління форми. Використання базових візуальних елементів управління. Компоненти форми для виключення помилкового введення даних користувачем. Використання візуальних елементів управління "дерево" та "таблиця". Робота з даними у XML-форматі.

7.2. Використання графічних можливостей платформи Microsoft .NET

Особливості графічного виведення даних. Логічна система координат. Простори імен GDI+. Обробка повідомлення перемальовування. Програмне генерування повідомлення перемальовування. Графічні об'єкти GDI+. Використання пензлів, пер та шрифтів. Робота з графічними зображеннями.

Тема 8. Розгортання програмного продукту

8.1. Структура модулю компіляції. Приватні та спільні модулі компіляції. Глобальний кеш модулів компіляції.

8.2. Створення спільних модулів компіляції. Загальні відомості про розгортання додатків. Види розгортання. Проекти установки та розгортання.

Тема 9. Системи баз даних. Основні поняття й архітектура

9.1. Основні поняття баз даних. Інфологічний та даталогічний рівні. Правило трьох "ні". Поняття "інформаційна система", "банк даних" та "база даних". Поняття СКБД, їх переваги та недоліки.

9.2. Архітектура баз даних. Концептуальний рівень. Зовнішній рівень. Внутрішній рівень.

Тема 10. Моделі даних

10.1. Моделювання даних.

Поняття про моделювання даних. Класифікація моделей. Ієрархічна модель даних. Ієрархічна структура даних. Операції над ієрархічною структурою, її переваги та недоліки.

Мережна модель даних. Мережна структура даних. Операції над мережною структурою. Переваги та недоліки мережної моделі.

10.2. Реляційна модель даних

Реляційна структура даних. Основні поняття та визначення.

Реляційна алгебра. Операції реляційної алгебри. Приклади застосування реляційної алгебри. Властивості операцій реляційної алгебри.

10.3. Семантичне моделювання предметної області

Модель "сутність – зв'язок". Поняття сутності, властивості, зв'язку. ER-моделювання предметної області. Типи ER-діаграм. Типи зв'язку на ER-діаграмах. Проектування бази даних за допомогою методу ER-моделювання.

10.4. Теорія нормалізації реляційної моделі даних

Аномалії при виконанні операцій у базі даних. Функціональні залежності. Аксиоматика функціональних залежностей. Логічне виведення функціональних залежностей. Аксиоми Армстронга. Декомпозиція без втрат. Теорема Хіта (Хеза). Визначення першої нормальної форми. Неповні функціональні залежності та друга нормальна форма. Транзитивні залежності й третя нормальна форма. Нормальна форма Бойса-Кодда. Багатозначні залежності. Теорема Фейгіна та четверта нормальна форма. Залежності за з'єднанням – п'ята нормальна форма. Загальна процедура нормалізації. Денормалізація відносин.

Тема 11. Мова SQL

11.1. Операції над схемою бази даних. Створення бази даних. Оператор CREATE DATABASE. Створення таблиці. Оператор CREATE TABLE. Модифікація таблиці. Оператор ALTER TABLE. Видалення таблиці. Оператор DROP TABLE. Видалення бази даних. Оператор DROP DATABASE.

11.2. Засоби пошуку даних. Основні конструкції мови, призначені для вибирання даних. Вирази, умови та оператори. Вибирання з кількох таблиць. Використання агрегатних функцій. Фраза GROUP BY, HAVING, ORDER BY. Підзапити.

11.3. Засоби маніпулювання даними. Додавання рядків до таблиці. Оператор INSERT. Оновлення даних. Оператор UPDATE. Видалення рядків таблиці. Оператор DELETE.

Тема 12. Проектування баз даних

12.1. Методологія проектування бази даних. Етапи проектування бази даних. Аналіз предметної області. Концептуальне моделювання предметної області. Логічне та фізичне проектування.

12.2. Функції групи адміністратора з проектування бази даних.

Тема 13. Транзакції та цілісність даних

13.1. Цілісність даних.

Поняття про обмеження цілісності. NULL-значення та тризначна логіка. Цілісність сутностей та зовнішніх ключів. Операції, що порушують посиальну цілісність. Стратегії підтримки посиальної цілісності.

Обмеження домену, атрибута, кортежу, відношення та бази даних. Обмеження цілісності у мові SQL.

13.2. Транзакції та цілісність даних. Проблеми паралельної роботи транзакцій

Поняття транзакції. Властивості транзакцій. Реалізація транзакцій засобами SQL.

Поняття суміші транзакцій. Проблеми паралельної роботи транзакцій. Проблема втрати результатів оновлення. Проблема незафіксованої залежності (читання "брудних" даних, неповторюване зчитування). Проблема несумісного аналізу.

Реалізація ізолюваності транзакцій засобами SQL.

Тема 14. Поняття та класифікація паралельних обчислювальних систем (ПОС). Класифікація ПОС за Флінном. Багатопроцесорні та багатокомпютерні системи. Принципи побудови ПОС зі спільною і розподіленою пам'яттю. Типи топологій комунікаційного зв'язку.

14.1. Поняття та класифікація паралельних обчислювальних систем (ПОС). Класифікація ПОС за Флінном. Багатопроцесорні та багатокомпютерні системи

Поняття паралельної обчислювальної системи (ПОС). Парадигми ПОС: одночасне виконання однієї або множини команд на однією та множиною областями пам'яті. Поняття багатопроцесорної та багатокомпютерної системи.

14.2. Принципи побудови ПОС зі спільною і розподіленою пам'яттю. Типи топологій комунікаційного зв'язку.

Поняття спільної та розподіленої пам'яті. Типи топологій зв'язку: кільце, решітка, зірка, повнозв'язаний граф, гіперкуб. Основні характеристики топології та їх розрахунок.

Тема 15. Послідовна та паралельні моделі програмування. Паралельні моделі програмування: паралелізм задач та паралелізм даних. Базо-

ві етапи розроблення паралельної програми (алгоритму). Багатопотокові програми.

15.1. Послідовна та паралельні моделі програмування. Паралельні моделі програмування: паралелізм задач та паралелізм даних.

Послідовна та паралельні моделі програмування.

15.2. Базові етапи розроблення паралельної програми (алгоритму). Багатопотокові програми.

Етапи розроблення паралельної програми: визначення глобальних та локальних змінних. Парадигма передачі повідомлень між фрагментами програми та результатами. Багатопоточність як засіб реалізації паралельної програми.

Поняття процесу та потоку. Засоби обміну даними в ПОС: передача повідомлень у системах з розподіленою пам'яттю і спільні змінні у системах зі спільною пам'яттю.

Тема 16. Графові моделі виконання паралельних програм

16.1. Графові моделі виконання паралельних програм (операцій-операнди).

Поняття та представлення графа для виконання паралельної програми. Склад дій, що відбувається у вершинах графа. Склад дій, що відбуваються у дугах графу. Приклади застосування моделі графу при виконанні простих програм паралельних обчислень.

Тема 17. Технології та мови паралельного програмування.

17.1. Особливості мов та технологій паралельного програмування як відображення парадигм на основі паралелізму даних та паралелізму завдань та використання спільної та розподіленої пам'яті.

Принципи побудови паралельних методів та алгоритмів: декомпозиція завдання на підзавдання; виокремлення інформаційних залежностей між підзавданнями; масштабування під завдань; розподіл під завдань між процесорами системи. Поняття паралелізму даних та паралелізму задач як технології роботи зі спільною та розподіленою пам'яттю.

Тема 18. Основи технології OpenMP при розпаралелювальні програм.

18.1. OpenMP - модель паралелізму з управління.

Принципи організації виконання OpenMP-програми. Модель Fork and Join. Модель пам'яті.

Тема 19. Програмні особливості розроблення OpenMP-програми.

19.1. Директиви, класи змінних OpenMP.

Основні поняття: директиви та клаузи. Поняття структурного блоку. Компіляція OpenMP-програм. Змінні оточення, які керують виконанням OpenMP-програм. Визначення загальних і локальних змінних, сумісна обробка локальних змінних (операція редукції).

19.2. Директиви для визначення паралельної області.

Директиви для розподілу обчислень всередині паралельної області: директиви `for`, `sections`, `single`. Розпаралелювання за циклами та за секціями. Засоби синхронізації, методи управління кількістю потоків в OpenMP.

Тема 20. Модель паралелізму на основі інтерфейсу передачі повідомлень MPI.

20.1. Основні принципи роботи MPI.

Парні (`point-to-point`) операцій між двома процесами і колективні (`collective`) комунікаційні дії для одночасної взаємодії декількох процесів. Поняття структури паралельної програми, операцій ініціалізації та завершення MPI-програми.

20.2. Основні функції MPI для підтримки передачі повідомлень.

Області зв'язку та комунікатори. Функції MPI підтримки двосторонніх та колективних операцій. Організація двосторонніх (точкових) взаємодій. Блокові паралельні алгоритми MPI. Призначення та склад колективних операцій передачі даних в MPI.

20.3. Функції підтримки виконання глобальних операцій в MPI.

Призначення та складу глобальних обчислювальних операцій над розподіленими даними; склад та призначення глобальних операцій редукції в MPI.

20.4. Управління комунікаторами.

Методи управління групами процесів і комунікаторами; операції щодо створення груп процесів та отримання інформації про групи процесів в MPI.

ПРИКЛАД ЗАВДАНЬ ЕКЗАМЕНАЦІЙНОГО БІЛЕТУ

Завдання 1.

.....

Створити клас `Student` (студент), розробивши наступні елементи класу:
Поля:

- прізвище;
- курс навчання;

Конструктори, що дозволяють створити екземпляр класу:

- з заданим прізвищем та курсом навчання;
- з заданим прізвищем для студента першого курсу.

Методи, що дозволяють:

- вивести прізвище студента на екран;
- розрахувати кількість років, що залишилися до закінчення студентом університету;

Властивості:

- одержати-установити курс навчання (доступне для читань і запису);
- що дозволяють отримати прізвище студента (доступне тільки для читання).

У клас `Student` додати індексатор, що дозволяє за індексом 0 звертатися до поля «прізвище», за індексом 1 - до поля «курс навчання», при інших значеннях індексу видається повідомлення про помилку.

У клас `Student` додати перевантаження:

- операцій `++` та `--`, які збільшують або зменшують значення поля «курс навчання», на 1;
- операції `>` та `<`, які дозволяють порівнювати два об'єкти класу за полем «курс навчання».

Розробити програму, в якій створити об'єкти створеного класу та продемонструвати роботу конструкторів, методів, властивостей, індексаторів та перевантажених операцій.

РОЗВ'ЯЗОК

```
class Student
{
    string name;
    int course;
// Конструктор з двома параметрами
    public Student(string name, int course)
    {
        this.name = name;
        this.course = course;
    }
}
```

```
// Конструктор з одним параметром
public Student (string name)
{
    this.name = name;
    this.course = 1;
}
// Метод, який виводить прізвище студента
public void PrintName()
{
    Console.WriteLine(name);
}
// Метод, який розраховує кількість років, що залишилися навчатися
public int GetYear()
{
    return 5- course;
}
// Властивості
public int Course
{
    set
    {
        course = value;
    }
    get
    {
        return course;
    }
}
public string Name
{
    get
    {
        return name;
    }
}
// Індексатори
public dynamic this [int index]
{
```



```

get
{
    if (index == 0)
        return name;
    else if (index == 1)
        return course;
    else
        return -1;
}
set
{
    if (index == 0)
        name = value;
    else if (index == 1)
        course = value;
    else
        Console.WriteLine("Error");
}
}
// Перевантаження операцій
public static Student operator ++ (Student s)
{
    s.course++;
    return s;
}
public static Student operator --(Student s)
{
    s.course--;
    return s;
}
public static bool operator >(Student s1, Student s2)
{
    if (s1.course > s2.course) return true;
    else return false;
}
public static bool operator <(Student s1, Student s2)
{
    if (s1.course < s2.course) return true;
}

```

```

        else return false;
    }

}

class Program
{
    static void Main(string[] args)
    {
        // Демонстрація роботи конструкторів
        Student s1 = new Student("Шевченко");
        Student s2 = new Student("Петренко", 4);
        // Демонстрація роботи методів
        Console.Write("Прізвище студента: ");
        s1.PrintName();
        Console.WriteLine("Залишилося навчатися {0} роки",s1.GetYear());
        // Демонстрація роботи властивостей
        s1.Course = 2;
        Console.WriteLine("Студент {0} перейшов на {1}
курс",s1.Name,s1.Course);
        // Демонстрація роботи індексаторів
        s1[0] = "Василенко";
        s1[1] = 3;
        Console.WriteLine(s1[0]);
        Console.WriteLine(s1[1]);
        // Демонстрація роботи перевантажених операцій
        s1++;
        Console.WriteLine(s1[1]);

        if (s1 > s2)
            Console.WriteLine("студент {0} старший, ніж студент {1}", s1.Name,
s2.Name);
        else
            Console.WriteLine("студент {0} молодший, ніж студент {1}", s1.Name,
s2.Name);
    }
}

```

Рис. 1.1 Код програми

Завдання 2.

База даних Хліб містить такі таблиці:

Товари (Код_товару, Товар, Ціна, Ціна_закупівлі).

Виробники (Код_виробника, Виробник, Адреса, Телефон).

Продажі (Код_продажу, Дата, Код_виробника, Код_товару, Кількість).

Для заданої бази даних необхідно:

1. Створити базу даних Хліб в СУБД MS SQL Server/MySQL.
2. Створити таблиці у базі даних Хліб та з'єднати їх.
3. Заповнити таблиці даними контрольного прикладу.
4. Самостійно сформулювати запит до бази даних та реалізувати його мовою SQL зі з'єднанням двох таблиць, з умовою фільтрування та впорядкуванням.
5. Самостійно сформулювати запит до бази даних та реалізувати його мовою SQL з групуванням та використанням агрегатних функцій.

Примітки. 1. Завдання виконуються у середовищі Visual Studio або Management Studio.

2. По кожному завданню зробити скріншоти з результатами виконання, а по завданнях 4 і 5 ще додатково записати формулювання задач та відповідні SQL-скрипти.

РОЗВ'ЯЗОК

1. Створити базу даних Хліб в СУБД MS SQL Server/MySQL.

Створено базу даних Хліб в СУБД MS SQL Server у середовищі Visual Studio (рис. 2.1).

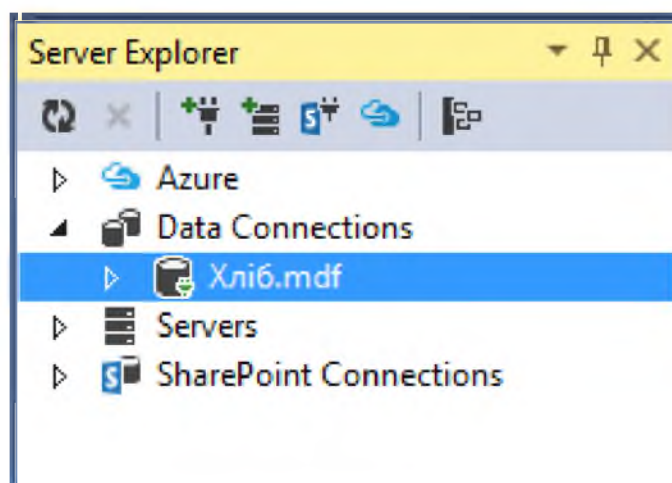


Рис. 2.1. База даних Хліб

2. Створити таблиці у базі даних Хліб та з'єднати їх.

Створено таблиці у базі даних Хліб (рис. 2.2) та встановлено зв'язки між ними (рис. 2.3).

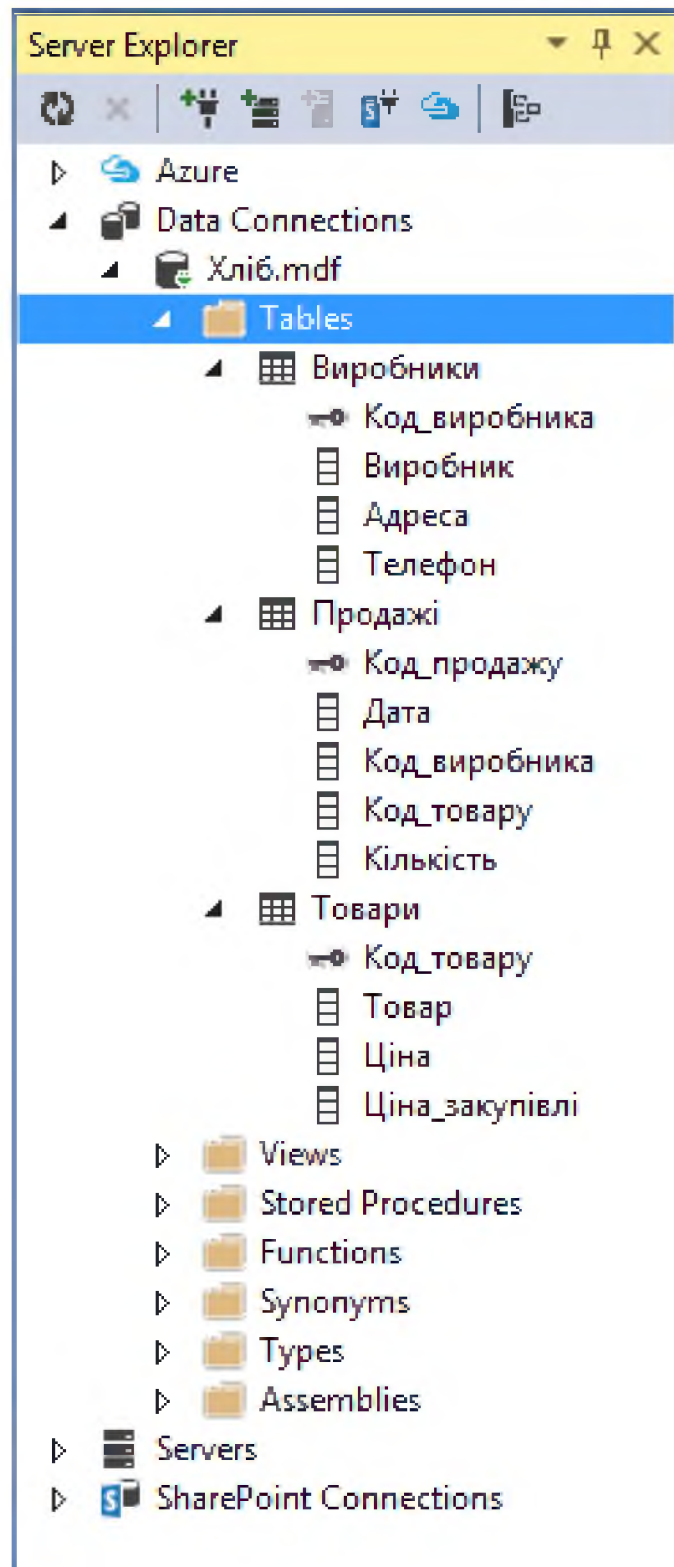


Рис. 2.2. Таблиці в базі даних Хліб

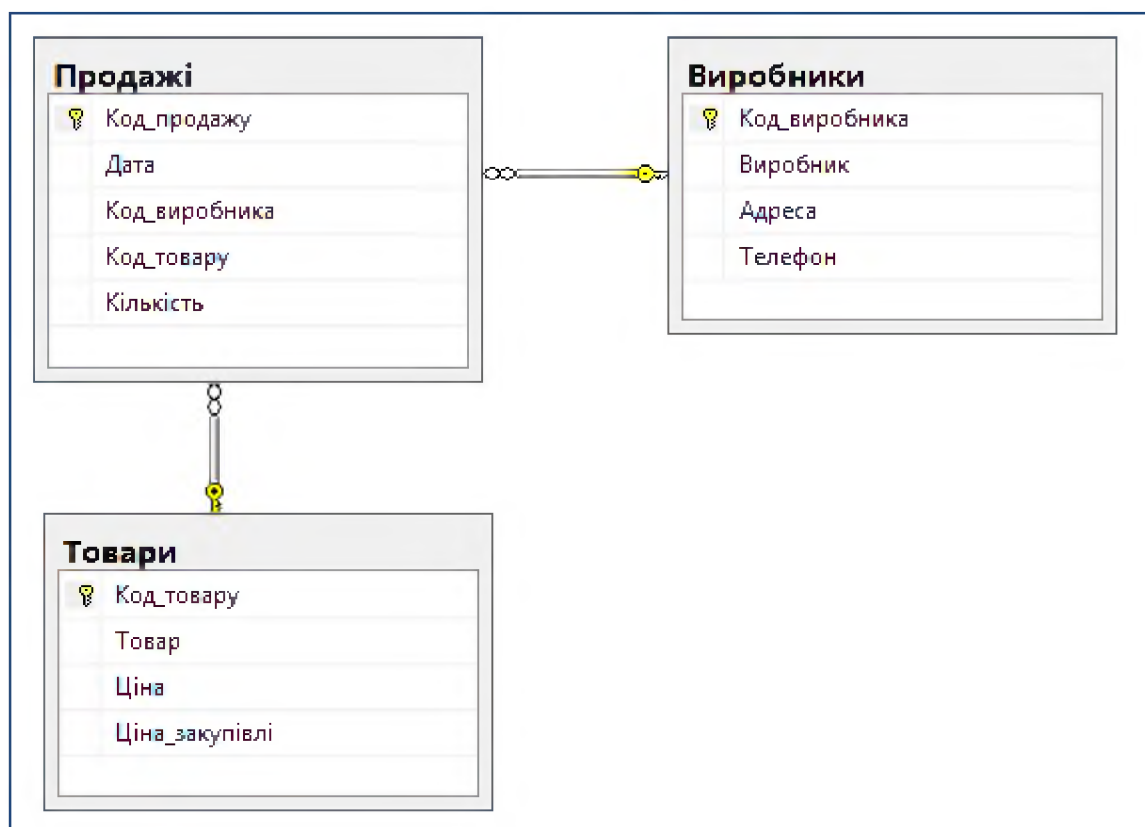


Рис. 2.3. Зв'язки між таблицями в базі даних Хліб

3. Заповнити таблиці даними контрольного прикладу.
Заповнено таблиці бази даних (рис. 2.4 – 2.6).

	Код_товару	Товар	Ціна	Ціна_закупівлі
▶	1	Хліб "Українськ...	13,0000	12,0000
	4	Батон "Молоч...	12,0000	11,0000
	5	Булка з маком	10,5000	10,0000
•	NULL	NULL	NULL	NULL

Рис. 2.4. Дані таблиці Товари

	Код_виробника	Виробник	Адреса	Телефон
▶	1	Х/з "Салтівськ...	м. Харків, вул. ...	(057)710-50-40
	2	Х/з "Кулиничі"	Харківський р-...	(0572)62-51-37
•	NULL	NULL	NULL	NULL

Рис. 2.5. Дані таблиці Виробники

	Код_продажу	Дата	Код_виробника	Код_товару	Кількість
▶	1	01.06.2017	1	1	200
	2	01.06.2017	1	4	250
	3	01.06.2017	2	1	150
	4	01.06.2017	2	5	180
	5	02.06.2017	1	1	220
	6	02.06.2017	1	5	170
	7	02.06.2017	2	1	200
	8	02.06.2017	2	4	100
•	NULL	NULL	NULL	NULL	NULL

Рис. 2.6. Дані таблиці «Продажі»

4. Самостійно сформулювати запит до бази даних та реалізувати його мовою SQL зі з'єднанням двох таблиць, з умовою фільтрування та впорядкуванням.

Завдання. Вивести такі дані про продажі товарів виробника з кодом 1: Дата, Товар, Ціна, Кількість, Вартість. Дані вивести за спаданням вартості.

SQL-скрипт.

```
SELECT Продажі.Дата, Товари.Товар, Товари.Ціна, Виробники.Виробник,
Продажі.Кількість, Товари.Ціна*Продажі.Кількість AS Вартість
FROM Товари JOIN Продажі ON Товари.Код_товару=Продажі.Код_товару
JOIN Виробники ON Виробники.Код_виробника=Продажі.Код_виробника
Where Виробники.Код_виробника=1
Order by Вартість Desc
```

Результат (рис. 2.7).

	Дата	Товар	Ціна	Виробник	Кількість	Вартість
1	2017-06-01	Батон "Молочний"	12,00	Х/з "Салтівський"	250	3000,00
2	2017-06-02	Хліб "Український"	13,00	Х/з "Салтівський"	220	2860,00
3	2017-06-01	Хліб "Український"	13,00	Х/з "Салтівський"	200	2600,00
4	2017-06-02	Булка з маком	10,50	Х/з "Салтівський"	170	1785,00

Рис. 2.7. Результат виконання завдання 4

5. Самостійно сформулювати запит до бази даних та реалізувати його мовою SQL з групуванням та використанням агрегатних функцій.

Завдання. Визначити кількість одиниць проданого товару кожного виду.

SQL-скрипт.

```
SELECT Товари.Товар, SUM(Продажі.Кількість) AS Загальна_кількість
FROM Товари JOIN Продажі ON Товари.Код_товару=Продажі.Код_товару
Group by Товари.Товар
```

Результат (рис. 2.8).

	Товар	Загальна_кількість
1	Батон "Молочний"	350
2	Булка з маком	350
3	Хліб "Український"	770

Рис. 2.8. Результат виконання завдання 5

Завдання 3.

Реалізувати на базі технології OpenMP рішення задачі:

знайти добуток двох матриць розміром 3 на 3 за допомогою розпаралелювання циклу:

А

1	2	3
4	5	6
7	8	9

Б

1	2	3
1	2	3
1	2	3

.....

РОЗВ'ЯЗАННЯ

1. Створення файлу скрита:

```
#!/bin/bash
#PBS -d /home/vc04/examination
#PBS -N
#PBS -l nodes=3:ppn=2
#PBS -V
```

Рис. 3.1. Файл скрита

2. Створення файлу програми розв'язку завдання.

```
int** b = new int *[3];
for(i=0;i<3;i++)
<----->b[i] = new int[3];
    b[0][0]=1;
    b[0][1]=2;
    b[0][2]=3;
    b[1][0]=1;
    b[1][1]=2;
    b[1][2]=3;
    b[2][0]=1;
    b[2][1]=2;
    b[2][2]=3;
....
int** c = new int *[3];
for(i=0;i<3;i++)
<----->c[i] = new int[3];
....
omp_set_num_threads(2);
#pragma omp parallel private(i,j)
{
<----->#pragma omp parallel for
<----->for(i=0;i<3;i++)
```

Рис. 3.2. Код програми


```
int** b = new int *[3];
for(i=0;i<3;i++)
<----->b[i] = new int[3];
    b[0][0]=1;
    b[0][1]=2;
    b[0][2]=3;
    b[1][0]=1;
    b[1][1]=2;
    b[1][2]=3;
    b[2][0]=1;
    b[2][1]=2;
    b[2][2]=3;
....
int** c = new int *[3];
for(i=0;i<3;i++)
<----->c[i] = new int[3];
....
omp_set_num_threads(2);
#pragma omp parallel private(i,j)
{
<----->#pragma omp parallel for
<----->for(i=0;i<3;i++)
```

Рис. 3.2. (продовження)

```

<----->     for (j=0;j<3;j++)
<----->     {
<-----><----->c[i][j]=0;
<-----><----->for (k=0;k<3;k++)
<-----><----->{
<-----><----->     c[i][j]+=a[i][k]*b[k][j];
<-----><----->}
<----->     }
        }
        for (i=0;i<3;i++)
        {
<----->for (j=0;j<3;j++)
<----->     cout << c[i][j] << " ";
<----->cout << endl;
        }
        for (i=0;i<3;i++)
        {
<----->delete[] a[i];
<----->delete[] b[i];
<----->delete[] c[i];
        }
        delete[] a;

```

Рис. 3.2. (продовження)

```

        delete[] b;
        delete[] c;
        return 0;
}..
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти

```

Рис. 3.2. (закінчення)

3. Компіляція додатку.

```

[vc04@cluster examination]$ g++
in -fopenmp
[vc04@cluster examination]$ █

```

Рис. 3.3. Компіляція додатку

4. Виконання – запуск завдання на обчислювальному кластері.

```
[vc04@cluster examination]$ qsub
16635.cluster
```

Рис. 3.4. Поміщення у чергу завдань на кластері та виконання завдання

```
6 12 18
15 30 45
24 48 72
[vc04@cluster examination]$
```

Рис. 3.5. Результат виконання завдання

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Бази даних у питаннях і відповідях : навчальний посібник / В. В. Чубук, Р. М. Чен, Л. А. Павленко та ін. – Х.: Вид. ХНЕУ, 2004. – 288 с.
2. Бейли Л. Изучаем SQL / Л. Бэйли. – СПб.: Питер, 2012. – 592 с.
3. Дейт Дж. Введение в системы баз данных / Дж. Дейт – 8-е изд. – М.: Вильямс, 2005. – 1328 с.
4. Дейт К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / Дж. Дейт. – СПб.: Символ-Плюс, 2010. – 480 с.,
5. Маки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов / А. Маки. – М. : "ИД Вильямс", 2010. – 416 с.
6. Гетьман О.О., Шаповал В.М. Економіка підприємства : навч. посіб. – 2-ге видання. – К. : Центр учбової літератури, 2010. – 488 с.
7. Нэш Т. С# 2010: ускоренный курс для профессионалов / Т. Неш.. – М. : "ИД Вильямс", 2010 – 592 с.
8. Об'єктно-орієнтоване програмування: конспект лекцій для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / Ю. Е. Парфьонов, В.М. Федорченко, М. Ю. Лосев та ін.. – Х. : Вид. ХНЕУ, 2010. – 312 с.
9. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Р. Максимчук, М. Энгл и др. – М. : "ИД Вильямс", 2008. – 720 с.
10. Тарасов О. В. Використання мови SQL для роботи з сучасними системами керування базами даних / О. В. Тарасов, В. В. Федько, М. Ю. Лосев. – Х. : Вид. ХНЕУ, 2013. – 348 с.

11. Тарасов О. В. Проективання баз даних : навч. посіб. / О. В. Тарасов, В. В. Федько, М. Ю. Лосєв. – Х. : Вид. ХНЕУ, 2011. – 200 с.
12. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен. – М. : "ИД Вильямс", 2013 – 1312 с.
13. Шилдт Г. C# 4.0: полное руководство / Г. Шилдт. – М. : "ИД Вильямс", 2011. – 1056 с.
14. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления / Воеводин В. В., Воеводин Вл. В. – СПб. : БХВ-Петербург, 2002. – 608 с.
15. Букатов А. А. Программирование многопроцессорных вычислительных систем / А. А. Букатов, В. Н. Дацюк, А. И. Жегуло. – Ростов-на-Дону : Издательство ООО «ЦВВР», 2003. – 208 с.
16. MPI: A Message-Passing Interface Standard Version 3.1. Метод доступа: <http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf>.
17. Корнеев В. Д. Параллельное программирование в MPI / В. Д. Корнеев. – Москва-Ижевск : "Институт компьютерных исследований", 2003. – 303 с.
18. Антонов А. С. Технология параллельное программирование MPI и OpenMP / А. С. Антонов. – М. : МГУ, 2000. – 344 с.
19. Антонов А. С. Параллельное программирование с использованием технологии MPI / А. С. Антонов. – М. : Изд-во МГУ, 2004. — 71 с.
20. Антонов А. С. Технология программирования MPI. Метод доступа: http://academy2012.hpc-russia.ru/files/lectures/mpi_openmp_track/mpi_c_academy4.pdf.
21. Абрамян М. Э. Практикум по параллельному программированию с использованием технологии MPI / М. Э. Абрамян. – Ростов-на-Дону, 2011. – 85 с.
22. Антонов А. С. Технологии параллельного программирования MPI и OpenMP : учебное пособие / А. С. Антонов. – М. : Издательство МГУ, 2012. – 344 с.
23. Антонов А.С. Параллельное программирование с использованием технологии OpenMP / А. С. Антонов. – М. : Изд. МГУ, 2009. – 77 с.
24. OpenMP Application Program Interface Version 4.0, July 2013.
Метод доступа: <http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf>.
25. Савельев В. А. Параллельное программирование: OpenMP API : методические указания / В. А. Савельев. – РГУ, 2006. – 32 с.
26. Немнюгин С. А., Стесик О. Л. Параллельное программирование для

многопроцессорных вычислительных систем / С. А. Немнюгин, О. Л. Стесик. – СПб. : БХВ, 2002. – 400 с.

27. Основы параллельного программирования с использованием технологий MPI и OpenMP : учебное пособие / Р. В. Жалнин, Е. Н. Панюшкина, Е. Е. Пескова, П. А. Шаманаев. – Саранск: Изд-во СВМО, 2013. – 78 с.

28. Левин М. П. Параллельное программирование с использованием OpenMP: курс лекций. Метод доступа :<http://www.intuit.ru/studies/courses/1112/232/info>).

29. Гергель В. П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем : учебное пособие / В. П. Гергель. – Нижний Новгород : изд-во ННГУ им. Н. И. Лобачевского, 2010. – 421 с.

Голова атестаційної комісії



Мінухін С. В.