

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

ЗАТВЕРДЖУЮ
Голова приймальної комісії
В. С. Пономаренко
2020р.



ПРОГРАМА

фахового вступного випробування
освітній ступінь «МАГІСТР»

спеціальність 126 «Інформаційні системи та технології»
освітньо-професійна програма «Інформаційні системи та технології»

Харків, 2020

Фахове випробування є комплексним іспитом, що включає практичні завдання (три завдання різного ступеню складності) з підготовки бакалаврів галузі знань 12 “Інформаційні технології”. До складу завдань внесені питання за наступними нормативними навчальними дисциплінами: об’єктно-орієнтоване програмування, організація баз даних і знань, технології комп’ютерного проектування.

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”

Спеціальні (фахові, предметні) компетентності
Здатність аналізувати об’єкт проектування та предметну область
Здатність до проектування системного, комунікаційного і прикладного програмного забезпечення, технічних засобів та комунікаційних й інформаційних технологій, мереж та систем.
Здатність розробляти, налагоджувати та вдосконалювати програмне забезпечення комп’ютерно-інтегрованих систем
Здатність використовувати сучасні технології проектування в розробці алгоритмічного та програмного забезпечення ІСТ
Здатність управляти якістю продуктів і сервісів ІСТ протягом їх життєвого циклу
Здатність розробляти та використовувати методи та математичні і комп’ютерні моделі фундаментальних і прикладних дисциплін для обробки, аналізу, синтезу та оптимізації результатів професійної діяльності, використовуючи методи формального опису систем

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	60	проектування бази даних предметної області
III. Рівень складності	–	1	1 евристичне завдання	70	технології комп'ютерного проектування

Практичні завдання включають завдання різного рівня складності, а саме одне діагностичне завдання та два евристичних завдання, загальна оцінка за виконання яких складає 200 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 мереж-

ною 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 посила- льну цілісність. Стратегії підтримки посилальної цілісності.

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

13.2. Транзакції та цілісність даних. Проблеми паралельної роботи транзакцій

Поняття транзакції. Властивості транзакцій. Реалізація транзакцій засо- бами SQL.

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

Реалізація ізолюваності транзакцій засобами SQL.

Тема 14. Технології комп'ютерного проектування та моделювання інформаційних систем

14.1. Склад та зміст принципів структурних методів та структурного аналізу для проектування ІС.

Основні ідеї структурних методів. Методи аналізу складних систем. По- няття "чорного ящика". Основні критерії функцій "чорного ящика". Принцип ієрархії як основний принцип розбудови складної системи на підсистеми. Побудова складної системи у вигляді ієрархії "чорних ящиків" та організація зв'язків між ними. Класифікація принципів структурного аналізу. Основні та неосновні принципи. Принципи декомпозиції та ієрархічного упорядкування. Сутність принципів ієрархії, декомпозиції, нотацій. Класифікація структур- них методологій та методи їх побудови.

14.2. Класифікація структурних методологій.

Методології SADT, структурного системного аналізу Гейна-Сарсона, структурного аналізу й проектування Йодана-Де Марко, розвитку систем Джексона, розвитку структурних систем Варньє-Орра, аналізу й проектуван- ня систем реального часу Уорда-Меллора і Хатлі, інформаційного моделювання Мартіна. Структурні карти Константайна та Джексона.

Поняття модуля, умовних і циклічних викликів, умовного та ітераційно- го вузла, структурного та процедурного блоків. Поняття проектної специфі-

кації, мініспецифікації. Методи побудови структурних методологій та їх порівняльний аналіз.

Тема 15. Концепція методології SADT та принципи побудови SADT-моделі й декомпозиції діаграм. Призначення стандарту IDEF0 та основних компонент діаграми

15.1. Основні елементи та правила побудови моделі SADT.

Склад елементів функціональної моделі SADT: функціональний блок та інтерфейсні дуги. Призначення інтерфейсних дуг. Деталізація діаграм на основі виявлення підфункцій та використання принципу декомпозиції. Поняття батьківської та "нащадкової" діаграм. Принцип спадкування властивостей (дуг) батьківських діаграм на діаграмах декомпозиції. Дерево діаграм декомпозиції.

15.2. Стандарт IDEF0 – розвиток методології SADT для проектування систем та структурно-функціонального моделювання процесів для автоматизації задач управління підприємством.

Поняття контекстної та декомпозованої діаграм стандарту IDEF0. Склад і призначення інтерфейсних дуг діаграм стандарту IDEF0. Принципи декомпозиції та правила використання та переносу інтерфейсних дуг блоків на діаграми нижчих рівнів. Призначення зворотних зв'язків з управління та входу та їх використання на діаграмах IDEF0. Нотація елементів на діаграмах IDEF0 різних рівнів декомпозиції (принцип спадкоємності).

Методи оцінювання збалансованості діаграм при їх декомпозиції. Кількісна оцінка процесу декомпозиції на основі показників збалансованості.

Тема 16. Стандарти опису потоків робіт та даних при застосуванні CASE-технологій

16.1. Призначення, нотації та структура діаграм потоків даних (DFD).

Нотації Йодана і Гейна-Сарсона для моделювання діаграм DFD. Особливості їх використання у потокових моделях. Склад і призначення основних елементів діаграм DFD: зовнішні сутності, процеси, системи (підсистеми), сховища даних, потоки даних.

16.2. Принципи побудови та декомпозиції діаграм DFD.

Нотація діаграм DFD. Побудова контекстної діаграми на основі визначення зв'язків системи, що моделюється, з зовнішнім середовищем. Етапи побудови діаграми. Принципи декомпозиції та побудови ієрархії діаграм DFD: одночасна декомпозиція даних та процесів, одноразове визначання однакових процесів, розміщення на одній діаграмі невеликої кількості елемен-

тів.

Тема 17. CASE-засоби аналізу та синтезу проектних рішень. Класифікація та характеристики CASE-засобів для автоматизації структурних методологій розроблення проекту ІС

17.1. Структура, функціональні особливості CASE-засобів.

Склад основних компонент інтегрованого CASE-пакета. Класифікаційні ознаки CASE-засобів. Узагальнююча характеристика CASE-засобів.

17.2. Характеристика програмних продуктів CASE-засобів.

Характеристика CASE-засобів: AllFusion Modeling Suite, Design/IDEF, EasyCASE, Vantage Team Builder, S-Designer, SILVERRAN, Visible Analyst, Oracle Designer/2000, Ramus Educational. Засоби функціонального, інформаційного та подвійного моделювання. Порівняльний аналіз CASE-засобів для проектування ІС.

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'ЯЗАННЯ

```
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'ЯЗАННЯ

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).

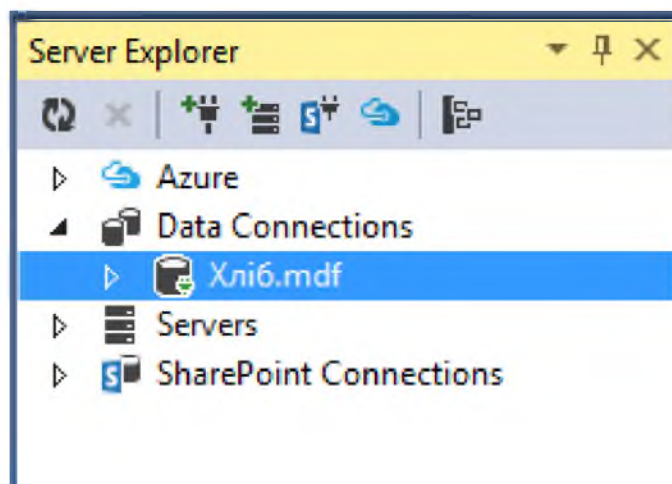


Рис. 2.1. База даних Хліб

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).

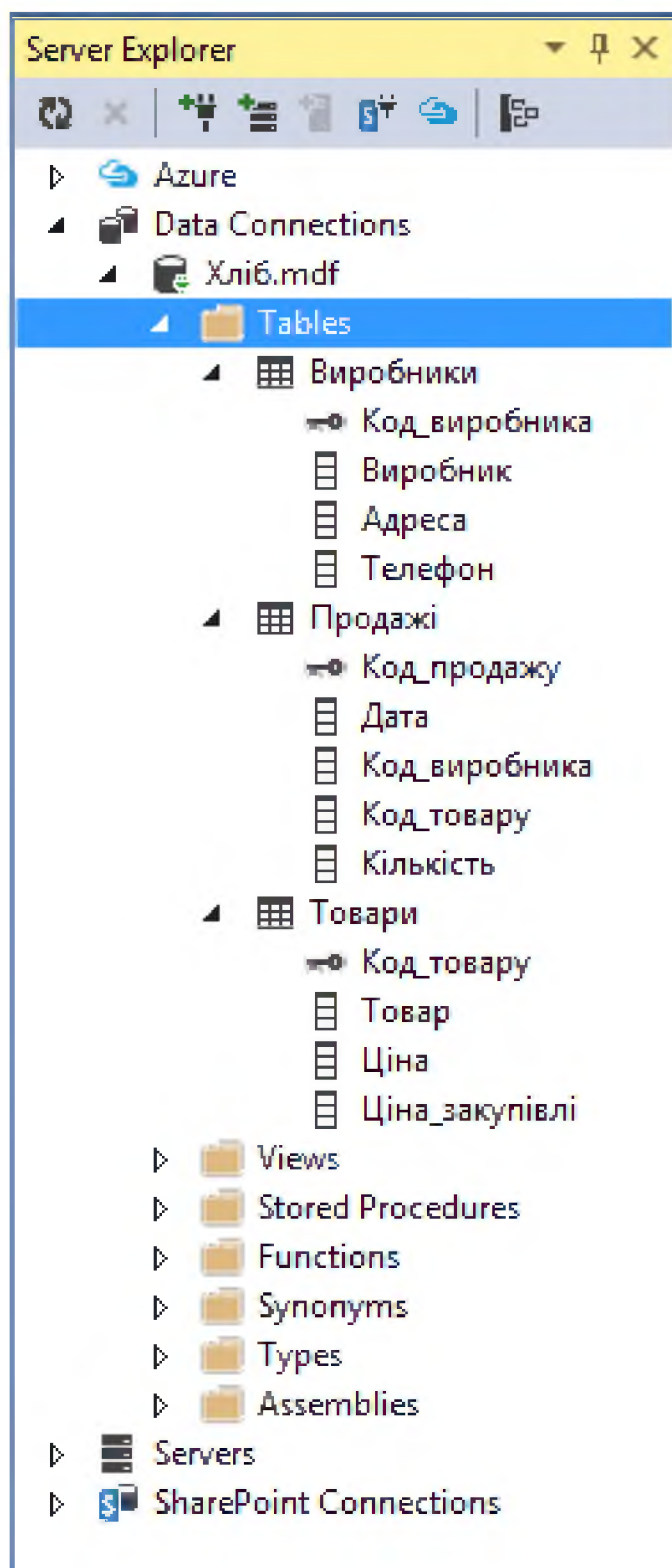


Рис. 2.2. Таблиці в базі даних Хліб

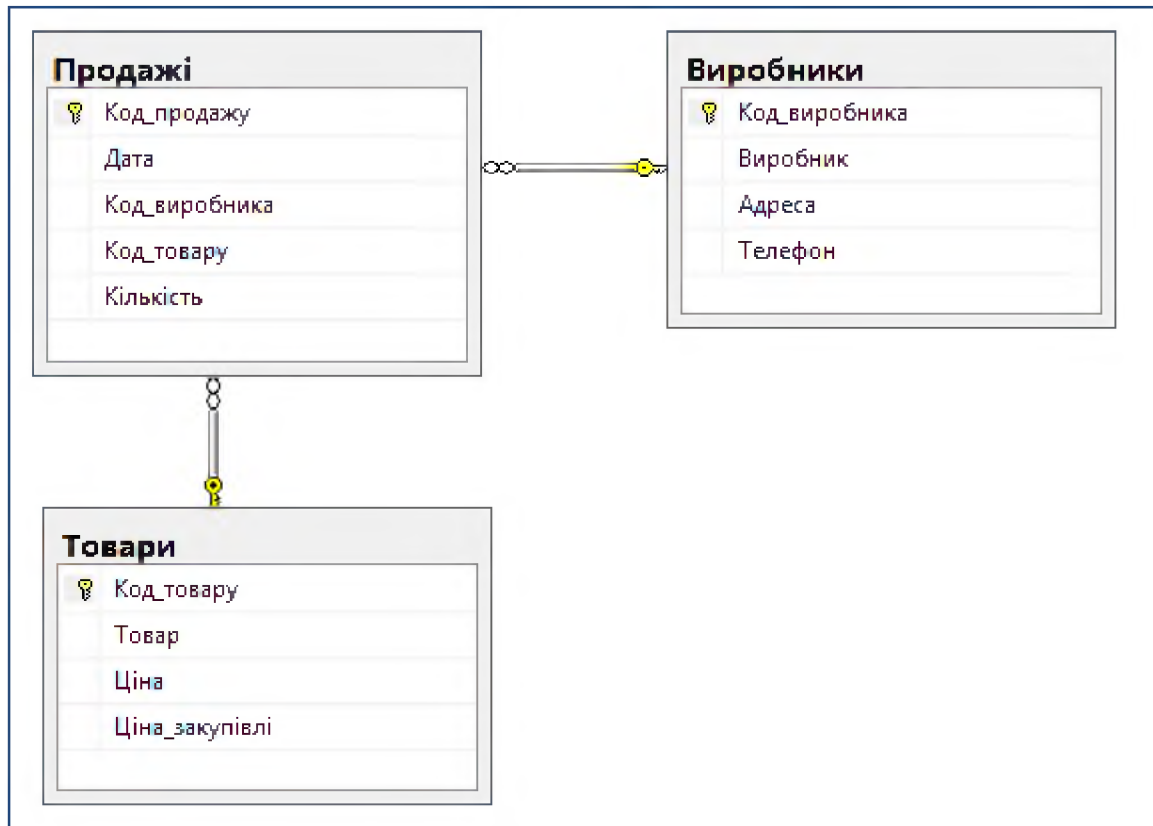


Рис. 2.3. Зв'язки між таблицями в базі даних Хліб

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.

Вхідна інформація для рішення завдання: замовлення, заявки; банківська виписка. Вихідна інформація: платіжна вимога; реєстр договорів; інформація про виконання договірних обов'язків. Процес рішення завдання регламентується: Нормативно-правовими актами договірних відносин; Формами типових договорів; Методикою бухгалтерського обліку; Методикою управлінського обліку. Виконавцями є комірник складу готової продукції; менеджер відділу збуту; економіст ПЕВ; головний диспетчер ПДВ; бухгалтер з фінансово-розрахункових операцій.

З метою розробки автоматизованої підсистеми управління договорами на сервері БД підприємства потрібно зберігати: дані про регіони; країни; клієнтів; ціни на продукцію; класифікатор продукції; стан договору; реєстр договорів; номенклатура продукції за договорами; план випуску продукції; рух продукції; план постачань.

Завдання:

побудувати контекстну діаграму рішення комплексу завдань у стандарті IDEF0;

розробити в стандарті IDEF0: діаграму A0 на основі наступних робіт: Укладання договору та моніторинг стану; Планування постачань; Планування та облік випуску продукції; Складський облік готової продукції та її відвантаження; Аналіз виконання зобов'язань за договорами;

діаграму A4 «Складський облік готової продукції та її відвантаження» на основі наступних робіт: Планування та облік відвантаження товару; Облік готової продукції на складі; Формування платіжних вимог.

побудувати контекстну діаграму рішення комплексу завдань у стандарті DFD;

розробити в стандарті DFD діаграму A0.

РОЗВ'ЯЗАННЯ

1. Побудувати контекстну модель бізнес-процесу за завданням у стандарті IDEF0 (рис. 3.1).

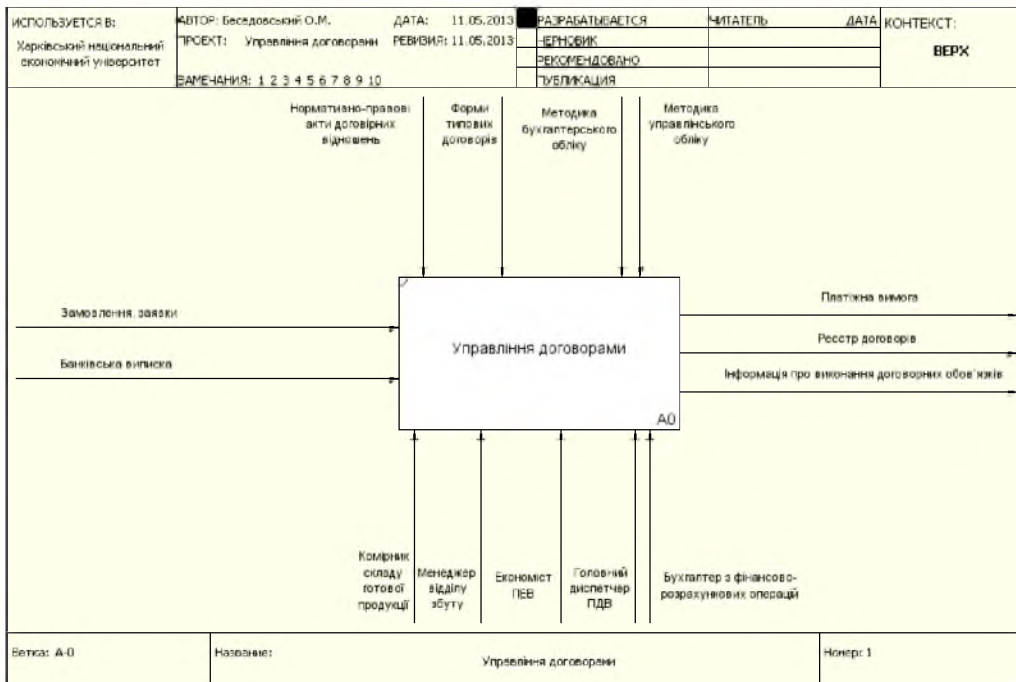


Рис. 3.1. Контекстна модель бізнес-процесу у стандарті IDEF0

2. Побудувати діаграму декомпозиції бізнес-процесу за завданням у стандарті IDEF0 (рис. 3.2).

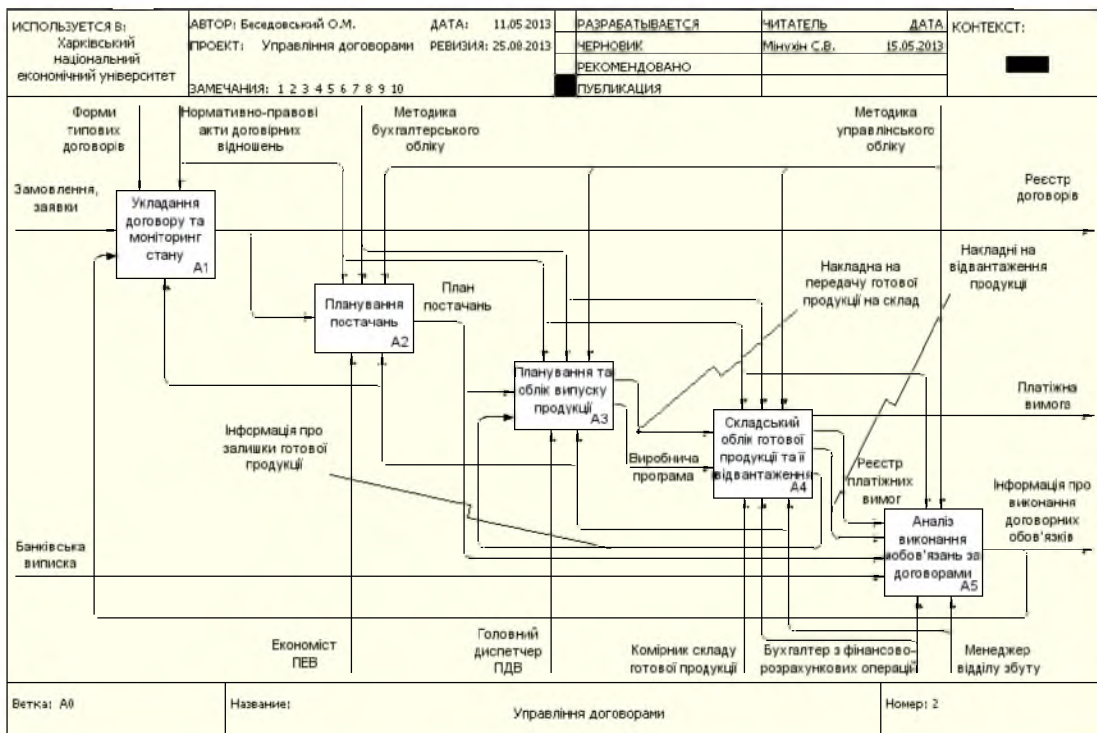


Рис. 3.2. Декомпозиція контекстної діаграми бізнес-процесу у стандарті IDEF0

3. Побудувати діаграму декомпозиції роботи А4 «Складський облік готової продукції та її відвантаження» (рис. 3.2).

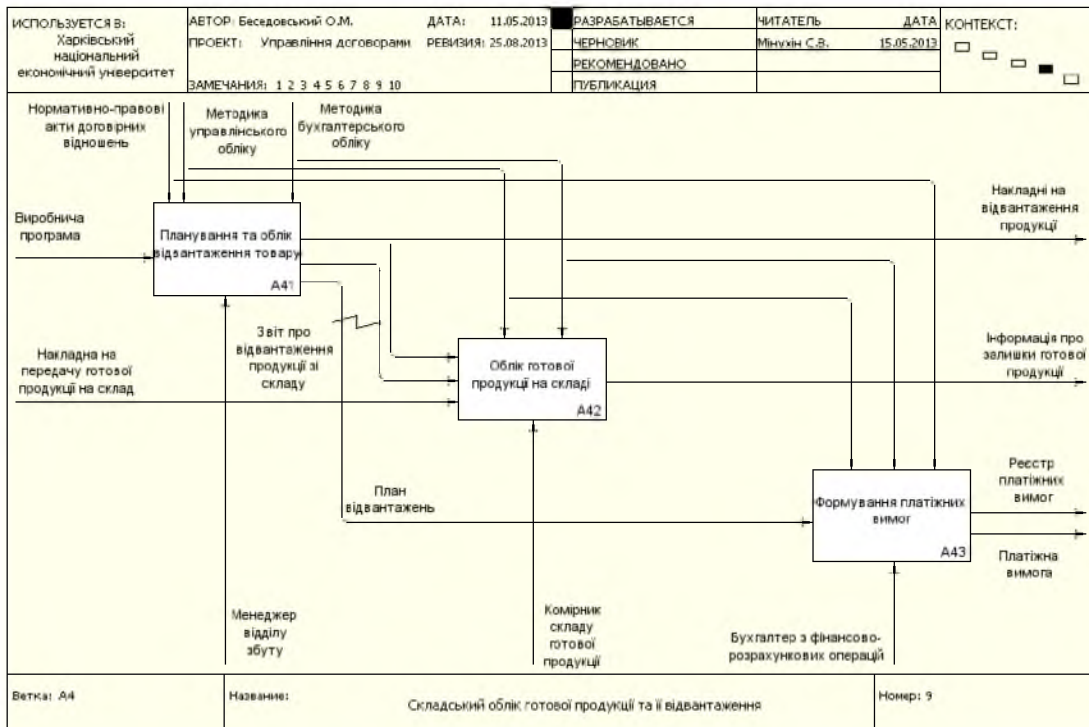


Рис. 3.2. Декомпозиція роботи «Складський облік готової продукції та її відвантаження»

4. Побудувати контекстну модель бізнес-процесу за завданням у стандарті DFD (рис. 3.4).

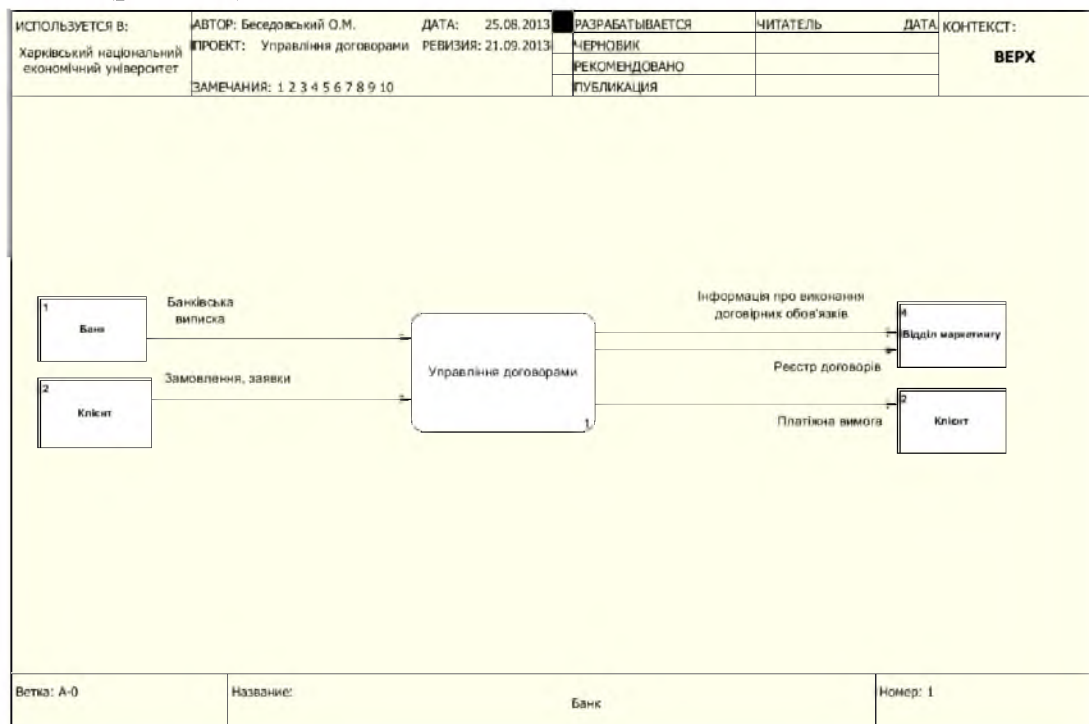


Рис. 3.4. Контекстна модель бізнес-процесу у стандарті DFD

5. Побудувати діаграму декомпозиції бізнес-процесу за завданням у стандарті DFD (рис. 3.5).

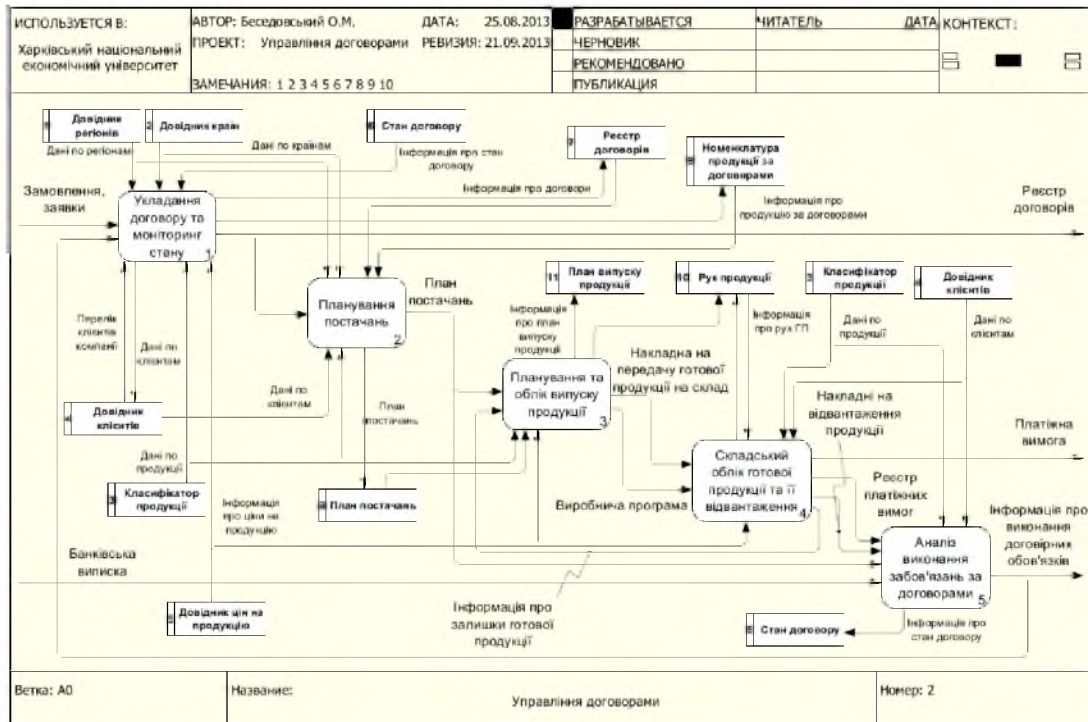


Рис. 3.5. Декомпозиція контекстної діаграми бізнес-процесу у стандарті DFD

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. Вендров А. М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров. – М. : Финансы и статистика, 2006. – 544 с.
4. Дейт Дж. Введение в системы баз данных / Дж. Дейт – 8-е изд. – М.: Вильямс, 2005. – 1328 с.
5. Дейт К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / Дж. Дейт. – СПб.: Символ-Плюс, 2010. – 480 с.,
6. Калашян А. Н. Структурные модели бизнеса: DFD-технологии / А. Н. Калашян, Г. Н. Калянов. – М. : Финансы и статистика, 2003. – 254 с.
7. Калянов Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов / Г. Н. Калянов. – 3-е изд. – М. : Горячая линия – Телеком, 2002. – 320 с.
8. Касьяненко В. О. Моделювання та прогнозування економічних процесів [Електронний ресурс] / В. О. Касьяненко. – Режим доступу :

<http://sumdu.telesweet.net/doc/lections/Modelyuvannya-ta-prognozuvannya-ekonomichnih-protsesiv/index.html>.

9. Лабораторний практикум з навчальної дисципліни "Технології комп'ютерного проектування" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання : / уклад. С. В. Мінухін, О. М. Беседовський, О. Б. Плоха. – Х. : ХНЕУ ім. С. Кузнеця, 2015. – 145 с.

10. Маки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов / А. Маки. – М. : "ИД Вильямс", 2010. – 416 с.

11. Мінухін С. В. CASE-технології. Конспект лекцій / С. В. Мінухін. – Х. : РВВ ХНЕУ, 2005. – 162 с.

12. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE-засобів / С. В. Мінухін, О. М. Беседовський, С. В. Знахур. – Х. : Вид ХНЕУ, 2008. – 272 с.

13. Моделирование бизнес-процессов / Электронный учебник [Электрон-ный ресурс]. – Режим доступа : <http://dit.isuct.ru/ivt/books/CASE/case10/index.htm>.

14. Нэш Т. С# 2010: ускоренный курс для профессионалов / Т. Неш.. – М. : "ИД Вильямс", 2010 – 592 с.

15. 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.

16. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Р. Максимчук, М. Энгл и др. – М. : "ИД Вильямс", 2008. – 720 с.

17. 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.

18. Тарасов О. В. Проектування баз даних : навч. посіб. / О. В. Тарасов, В. В. Федько, М. Ю. Лосев. – Х. : Вид. ХНЕУ, 2011. – 200 с.

19. Троелсен Э. Язык программирования С# 5.0 и платформа .NET 4.5 / Э. Троелсен. – М. : "ИД Вильямс", 2013 – 1312 с.

20. Шилдт Г. С# 4.0: полное руководство / Г. Шилдт – М. : "ИД Вильямс", 2011. – 1056 с.