

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Державний вищий навчальний заклад  
«Миколаївський політехнічний коледж»

**Методичні рекомендації щодо виконання та захисту  
курскових проектів  
з дисципліни «Інструментальні засоби візуального  
програмування»**

Укладач: Плоткіна О.Г. ст викладач вищої категорії  
Новожилова О.В. викладач першої категорії

Розглянуто і затверджено на засіданні циклової комісії «Інженерія  
програмного забезпечення»  
відділення «Економіка, телекомунікації та інформаційні технології»

Протокол № 4 від 14.11. 2017 р.

Голова циклової комісії \_\_\_\_\_ Плоткіна О.Г.

Методичні рекомендації щодо виконання та захисту курсових проектів з дисципліни «Інструментальні засоби візуального програмування» - 34 с.

Розглянуто і схвалено на семінарі методоб'єднання викладачів інформатики та програмування ВНЗ I-II р.а. Миколаївської області

Протокол № \_\_\_ від \_\_\_\_\_ 20\_\_ р.

Голова обласного методичного об'єднання

Круковська В.О. ,

Голова циклової комісії 0501 напряму Інформатика та ОТ, викладач МБК КНУБА

Рецензент: доцент кафедри комп'ютерної інженерії  
МНУ ім. В.О. Сухомлинського,  
к.т.н. Кузьма. К.Т \_\_\_\_\_

Методичні рекомендації до виконання курсового проекту містять відомості щодо: тематики, організації процесу проектування, методики виконання розділів проекту та опису результатів, вимог до програмної складової проекту. Наведено рекомендації по вибору інструментальних засобів конструювання програмного забезпечення, вимоги щодо оформлення матеріалів проекту, перелік основної і додаткової літератури, критерії оцінювання курсового проекту.

Методичні рекомендації створено для практичного використання студентами та викладачами спеціальності «Інженерія програмного забезпечення» при виконанні курсового проектування з дисципліни «Інструментальні засоби візуального програмування».

Розглянуто і схвалено на засіданні методичної ради  
Миколаївського політехнічного коледжу

Протокол № \_\_\_ від \_\_\_\_\_ 20\_\_ р.

Голова методичної ради коледжу \_\_\_\_\_

## ЗМІСТ

Вступ	4
1 Тематика курсових проєктів	5
2 Організаційне забезпечення курсового проєктування	5
3 Мета і задачі курсового проєкту	6
4 Послідовність виконання курсового проєкту	7
5 Опис результатів проєктування	8
5.1 Методичні рекомендації щодо змісту розділів проєкту	8
5.2 Вимоги до змісту розділу «Вступ»	8
5.3 Розділ «Загальна частина»	9
5.3.1 Опис предметної області	9
5.3.2 Формування вимог до програмного забезпечення задачі	10
5.3.3 Технічне завдання	12
6. Розробка проєкту програми	12
6.1 Вимоги до програмної складової	12
6.2 Зміст розділу «Ескізний проєкт»	13
6.3 Зміст розділу «Технічний проєкт»	14
6.4 Зміст розділу «Робочий проєкт»	16
7 Склад програмної документації	19
8 Порядок захисту курсового проєкту	19
9 Критерії оцінювання курсового проєкту	20
10 Вимоги стандартів до оформлення пояснювальної записки	22
11 Список літературних джерел	25
Додатки	28
Додаток 1 Орієнтований перелік тем	28
Додаток 2 Приклад завдання на курсовий проєкт	29
Додаток 3 Приклад діаграм потоків даних	30
Додаток 4 Приклад концептуальної моделі БД	32
Додаток 5 Приклад діаграми станів програми	33
Додаток 6 Приклад розділу «Програма та методика тестування»	34
Додаток 7 Приклад презентації	39

## ВСТУП

Курсовий проект з дисципліни «Інструментальні засоби візуального програмування» виконується студентами спеціальності 121 «Інженерія програмного забезпечення» на завершальному етапі навчання і повинен бути завершеною розробкою програмного застосування виконаною за вимогами програмної інженерії та діючих в Україні стандартів [1-4].

Подана методологія проектування вимагає дотримання основних етапів класичного життєвого циклу програмних продуктів: дослідження предметної області, формування вимог, проектування, конструювання, тестування та налагодження, документування програмного продукту.

Технологія роботи над проектом націлена на закріплення, поглиблення і узагальнення отриманих студентами теоретичних знань та практичних навичок при вирішенні конкретного фахового завдання.

Вибрані методи та засоби розробки програмного забезпечення відповідають сучасному рівню розвитку інструментальних засобів програмування, досягнень науки і техніки, вимог до програмної документації та ергономічних вимог щодо інтерфейсу користувача. Технологічними інструментами розробки є інтегроване середовище візуального програмування C++Builder, Use Case засоби проектування програмного забезпечення та системи керування базами даних.

Метою методичних рекомендацій є допомога студентам у процесі виконання курсового проекту.

Вирішення проблем і питань, поставлених в завданні на курсове проектування, сприяє розвиненню у студентів компетенцій у сфері науково-практичного пошуку, оволодінню методикою дослідження й експериментування, розвитку творчих здібностей, набуттю навичок формування програмної документації.

## **1 Тематика курсових проектів**

Тематика курсових проектів відповідає завданням навчальної дисципліни і тісно пов'язана з практичним застосуванням інструментальних засобів візуального програмування. Теми курсових проектів щорічно коригуються з урахуванням набутого в комісії досвіду, побажань спеціалістів, які беруть участь у рецензуванні проектів, рекомендацій Державної кваліфікаційної комісії (ДКК) і потребами навчального закладу. Затверджуються теми рішенням циклової комісії. Орієнтований перелік тем курсового проекту наведено в додатку 1.

Тематика направлена на проектування бази даних та програмного додатку до неї для реалізації задач інформаційної системи заданої предметної області (ПрО), з використанням засобів системи керування базами даних (СКБД) та інструментальних засобів візуального програмування (ІЗВП). Низка тем зорієнтована на вирішення практичних питань впровадження інформаційних технологій у навчальний процес.

Студенту надається право вибирати тему курсового проекту з переліку затверджених, або запропонувати свою тему з обґрунтуванням необхідності її розробки.

## **2 Організаційне забезпечення курсового проектування**

Завдання на курсове проектування (додаток 2) видається студенту після двох тижнів навчання з початку семестру .

Керівництво курсовим проектам здійснюється, як правило, викладачем дисципліни. На нього покладено відповідальність за організацію і контроль процесу виконання курсового проекту. Він складає графік виконання курсового проекту, консультує студента по всім питанням, контролює хід виконання робіт.

Приступаючи до роботи над курсовим проектом, студенту необхідно осмислити проблему, творчо й критично її дослідити. Рекомендовано дотримуватися наступного алгоритму: визначення мети і завдань проекту; виявлення і підбір літератури з теми, її вивчення; дослідження предметної області автоматизації; визначення вимог до програмного продукту; формування технічного завдання на розробку; проектування бази даних і програмного додатку до неї; тестування та налагодження програмного додатку; формулювання висновків і рекомендацій; документування ПЗ; написання вступу; оформлення списку використаних літературних джерел та додатків; літературне й технічне оформлення проекту; рецензування проекту; підготовка до

захисту і захист курсового проекту.

Курсовий проект має свою специфіку, його деталі завжди потрібно узгоджувати з керівником. Студент зобов'язаний після кожної фази проектування представляти керівнику отримані результати. Керівник перевіряє виконану роботу, вказує на помилки, роз'яснює недопрацьовані місця та дає рекомендації по їх виправленню.

### **3 Мета і задачі курсового проекту**

Метою курсового проектування є оволодіння методами творчого вирішення питань дослідження ПрО, формування вимог до майбутнього програмного продукту, проектування бази даних та програмного додатку до неї, який забезпечує доступ до інформації її обробку та формування результатів.

Основним завданням курсового проекту є систематизація студентами отриманих теоретичних та практичних знань зі спеціальних дисциплін з програмування: “Основи програмування та алгоритмічні мови”, “Бази даних”, “Операційні системи”, “Основи програмної інженерії”, “Конструювання програмного забезпечення», “Інструментальні засоби візуального програмування».

Об'єктом для курсового проектування є операції по збору, накопиченню та обробленню інформації в конкретній предметній області (підрозділи підприємств, установ, організацій різних галузей діяльності).

Студент повинен проявити вміння з використання методів і інструментальних засобів сучасних технологій розробки програмних продуктів, вміння користуватися спеціальною літературою, державними стандартами, довідниками та іншими матеріалами з інформаційних технологій.

Виконуючи розробку курсового проекту студент проходить всі етапи процесу створення інформаційної системи та програмного забезпечення для реалізації її задач. Йому необхідно: зібрати і систематизувати літературні джерела; провести дослідження інформаційних потоків та технології обробки інформації в ПрО; визначити перелік операцій, які підлягають автоматизації; сформулювати висновки, пропозиції і рекомендації з предмету дослідження. Результати досліджень - основа при вирішенні практичних завдань по конструюванні БД та розробки ПЗ з дотриманням вимог стандартів та основних положень програмної інженерії.

Опис результатів розробки наводиться в пояснювальній записці до курсового проекту. Складові пояснювальної записки наступні: вступ, загальний розділ,

спеціальний розділ, перелік використаної літератури та висновки. Розділи пояснювальної записки повинні бути логічно пов'язані. Програмний додаток створюється з урахуванням вимог ПП щодо якості програмного коду, повної реалізації запланованих функцій, використання нових технологій і інструментальних засобів.

#### **4 Послідовність виконання курсового проекту.**

Після отримання завдання на курсове проектування студент разом із керівником визначає: межі розкриття вибраної теми, предметну область для якої буде розроблюватися програмне забезпечення та приступає до її дослідження.

При з'ясуванні об'єкта, предмета і мети дослідження необхідно враховувати те, що між ними і темою курсового проекту є системні логічні зв'язки.

Результати досліджень, проектування, тестування та налагодження програмного додатку та ін. описуються у відповідних розділах пояснювальної записки. Завершальна частина пояснювальної записки - висновки щодо ефективності розробленого програмного продукту.

Рекомендується наступна послідовність дій:

дослідження ПрО, обробка результатів та узагальнення їх в описі предметної області;

формування вимог до програмного продукту;

дослідження інформаційних потоків та проектування БД;

використання інструментальних засобів при проектування інформаційного забезпечення задачі та архітектури програмного забезпечення;

специфікації програмних модулів та розробка їх алгоритмів;

конструювання програмного забезпечення;

програмна реалізація алгоритмів з використанням ІЗВП;

налагодження та тестування програмного виробу;

опис результатів;

подання розробки таким чином, щоб нею могли користуватися інші;

підготовка програмної документації;

створення презентаційних матеріали для захисту проекту.

## **5 Опис результатів проектування**

### **5.1 Методичні рекомендації щодо змісту розділів проекту**

Готуючись до викладення тексту курсового проекту, доцільно ще раз уважно прочитати назву теми, що містить проблему, яка повинна бути розкрита. Проаналізований та систематизований матеріал викладається відповідно до змісту у вигляді окремих розділів і підрозділів. Кожний розділ висвітлює самостійне питання, а підрозділ - окрему частину цього питання.

Тема має бути розкрита без пропуску логічних ланок, тому починаючи працювати над розділом, потрібно постійно дотримуватися його головної ідеї, а також призначення кожного підрозділу. Викладки розділів повинні підтверджуватися фактами, отриманими при виконанні робіт відповідного етапу (діаграми, схеми, фрагменти програмного коду, екранні форми, тощо), аналізом конкретного практичного результату. Треба уникати безсистемного викладення фактів без достатнього їх осмислення та узагальнення.

Описи робіт над проектом мають бути логічно пов'язані між собою. Один висновок не повинен суперечити іншому, а підкріплювати його. Якщо висновки не будуть пов'язані між собою, текст втратить свою єдність. Послідовність розділів повинна відповідати основним етапам розробки програмного забезпечення.

### **5.2 Вимоги до змісту розділу «Вступ»**

У вступі: обґрунтовується актуальність теми, що розроблюється та її практична значущість; визначаються об'єкт, предмет, мета і завдання розробки; наводяться методи, за допомогою яких виконувалася розробка; розкривається структура проекту та його основний зміст.

Обов'язковою частиною вступу є систематизація результатів вивчення матеріалів, із літературних джерел чи інтернет – ресурсів із практики вирішення поставленої проблеми, виявлення її типового стану, типових недоліків і труднощів та їх причин, типових результатів передового досвіду з вирішення досліджуваної проблеми.

Якщо, на момент виконання розробки, існують аналогічні програмні засоби, які реалізують автоматизацію процесів обробки інформації в ПРО, то необхідно вказати в чому їх переваги та недоліки. Якщо розробка проекту проводиться в межах вже існуючої інформаційної системи, то вказується, яке розширення функцій діючої



системи забезпечує розробка, в чому полягає її новизна та необхідність.

У вступі також висвітлюється теоретична та практична значущість роботи.

Якщо студент вирішив не торкатися деяких аспектів теми, він повинен зазначити про це у вступі.

Вступ доцільно писати після того, як написана основна частина курсового проекту.

### **5.3 Розділ «Загальна частина»**

#### **5.3.1 Опис предметної області**

Для розкриття результатів дослідження ПрО необхідно врахувати наступні положення:

1. Мета дослідження пов'язана з об'єктом і предметом дослідження, а також з його кінцевим результатом і методом його досягнення.

2. Необхідно чітко усвідомити образ очікуваного результату. Кінцевий результат дослідження повинен відобразити стан проблемної ситуації, перелік «тонких місць» в технологічному ланцюгу збору, накопичення та оброблення інформації та підтвердити необхідність автоматизації цих операцій.

3. Виконання розробки повинно забезпечити підвищення ефективності управлінських рішень і як результат, більш ефективне функціонування об'єкту дослідження ( суспільна і конкретна корисність щодо предмета дослідження).

При дослідженні ПрО необхідно визначити і вивчити її основні задачі, організаційну структуру, існуючу сукупність відношень, функціонуючі потоки інформації, джерела виникнення та джерела споживання інформації.

Особливо важливо досконально дослідити технологію обробки інформаційних потоків, визначити перелік найбільш рутинних та об'ємних технологічних операцій обробки даних.

Визначаючи об'єкт, треба знайти відповідь на запитання: що розглядається?, що необхідно автоматизувати? Об'єкт дослідження - це частина об'єктивної реальності, яка є предметом практичної і теоретичної діяльності студента.

За результатами досліджень ПрО виділяються інформаційні потоки (які документи оброблюються), функції збору та обробки інформації (яка технологія обробки документів), види звітів, перелік критеріїв та структури запитів (що треба отримати в результаті обробки документів). З отриманих даних, на наступних етапах,

виокремлюються операції проектування бази даних, операції зі збору, накопичення та обробки інформації, формуються функціональні вимоги до програмного продукту, визначається перелік запитів та звітів. Рекомендується наступний склад розділу «Опис предметної області»:

- організаційна схема;

- основні задачі підрозділу;

- опис рішень, що приймаються в предметній області, яка інформація необхідна для прийняття обґрунтованих рішень, їх періодичність і тривалість, обмеження та критерії прийняття рішень;

- економічний зміст задач, при вирішенні яких приймаються рішення для управління об'єктами предметної області;

- інформація (внутрішня та зовнішня), яка необхідна для прийняття обґрунтованих рішень, типи даних, джерела їх надходження, підходи та методології до їх збирання та оброблення.

В завершення підрозділу, наводяться основні фактори, які вказують на необхідність автоматизації технологічних операцій оброблення інформації, створення внутрішньомашинної бази даних та програмного додатку до неї.

### **5.3.2 Формування вимог до програмного забезпечення задач**

Етап формування вимог до програмного забезпечення задач є визначальним при забезпеченні його якості, термінів та вартості робіт. Саме на цій фазі мають бути зафіксовані реальні потреби користувачів, що стосуються функціональних, операційних та сервісних можливостей які планується реалізувати в розробці. Ціна помилок та нечітких неоднозначних формулювань на цій фазі дуже висока, бо час і засоби витрачаються на непотрібну програму. Відсоток помилок у постановці завдань перевищує відсоток помилок кодування, і це є наслідком суб'єктивного характеру процесу формулювання вимог. Для успішного формулювання вимог рекомендується скористуватися доступними інструментальними засобами проектування ПЗ, наприклад, UML Rational Rose для створення діаграма прецедентів (Use Case – діаграма). На діаграмі у наочній формі подається перелік основних користувачів (акторів) майбутнього ПЗ та функції (варіанти використання, прецеденти), які вони очікують від системи. Приклад діаграми прецедентів наведено на рисунку 1.

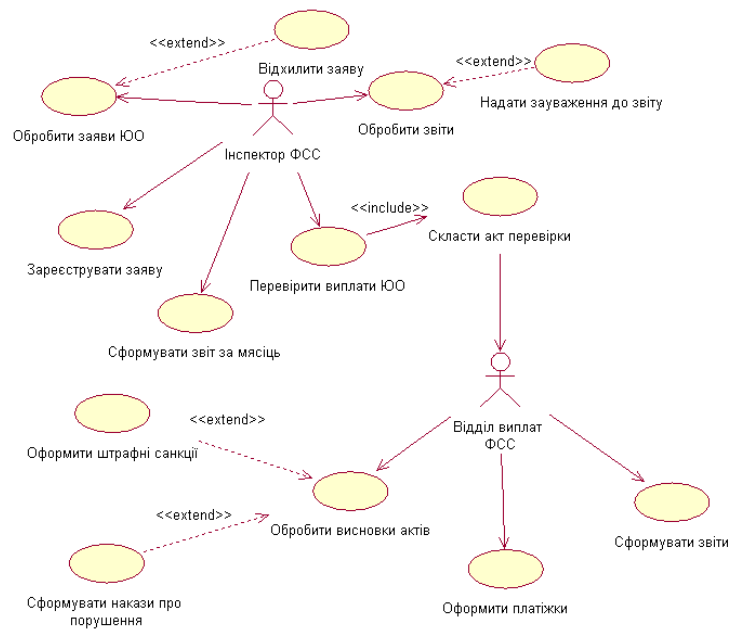


Рисунок 1 – Приклад діаграми прецедентів

В реальних умовах діаграми прецедентів дозволяють подати Замовнику в наочній формі користувачів та функції майбутнього ПЗ, які вони активізують. При обговоренні вимог до програмної розробки наявність таких діаграм сприяє порозумінню між Замовником і Виконавцем в питаннях уточнення чи розширення вимог.

Кожен прецедент повинен бути описаний за такою структурою:

Заголовок.

Передумови.

Головний потік.

Підпотоки.

Альтернативні потоки

Постумови.

Опис прецедента наведено в Прикладі 1.

*Приклад 1* Опис прецедента «Обробити заяви юридичних осіб»

Заголовок. Прецедент «Обробити заяви ЮО»

*Передумови.* Наявність реєстраційного номеру заяви юридичної особи в БД

*Головний потік.* Прецедент починає виконуватися, коли інспектор підключається до системи. Система надає йому форму для введення даних заяви ЮО. Введені дані перевіряються на повноту і коректність і, в разі відповідності цим вимогам, записуються в БД.

*Підпотоки:* «Визначити обсяги заявлених коштів за напрямками», «Отримати дані заяв вказаної ЮО»

*Альтернативні потоки.* Якщо дані не повні, то система формує відповідне повідомлення користувачу. Заява відхиляється. Транзакція БД не відбувається.

*Постумови.* В результаті виконання прецеденту дані заяви заносяться в БД. Підпотоки «Визначити обсяги заявлених коштів за напрямками», «Отримати дані заяв вказаної ЮО» формують добір даних з БД за вказаними критеріями пошуку та відображають їх у екранній формі.

### **5.3.3 Технічне завдання**

В технічному завданні (ТЗ) закріплюються функції, які повинні бути реалізовані в програмному додатку, перелік робіт, терміни їх виконання, технічне забезпечення, тощо. ТЗ на розробку регламентується ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

Основні розділи ТЗ:

Найменування об'єкту розробки, та область застосування.

Мета розробки.

Джерела розробки.

Технічні вимоги.

Економічні показники.

Порядок контролю і приймання об'єкту.

## **6. Розробка проекту програми**

### **6.1 Вимоги до програмної складової**

Результатом виконання курсового проекту повинна бути БД та повноцінний програмний додаток до неї, який реалізує задачу інформаційної системи конкретної ПрО. Розробка повинна виконуватися за вимогами програмної інженерії з використанням певних інструментальних засобів візуального програмування (C++ Builder, Borland Delphi, мови HTML, PHP, C#, Case-засобів, тощо). Програмний додаток повинен надавати користувачу можливості здійснення операцій формування та ведення бази даних, обробки даних з метою формування звітів, чи добору інформації за визначеними критеріями.

Для отримання означених характеристик необхідно дотримуватись наступних вимог:

1. Розробка програмного проекту повинна бути здійснена на основі глибокого і ретельного аналізу предметної області, літературних джерел з вибраної теми.

Можливість використання та доцільність розробки повинна бути всебічно обґрунтованими. В програмному проекті повинні бути реалізовані всі функції, які визначалися на основі аналізу вимоги, які формувалися на першому етапі.

2. Результати роботи програми повинні бути протестовані на реальних прикладах з використанням класів еквівалентності. Інтерфейс користувача повинен бути зрозумілим і таким, що легко засвоюється. База даних повинна бути захищеною від непередуманих дій користувача та несанкціонованого доступу.

3. Графічна частина розробки повинна подаватися у тексті пояснювальної записки до курсового проекту у вигляді відповідних рисунків – діаграм, а саме: діаграми варіантів використання, діаграми класів, діаграми станів програми, діаграми потоків даних, взаємодії модулів програми тощо. При цьому всі схеми та рисунки повинні бути оформлені відповідно до прийнятих правил оформлення технічної документації.

4. При програмній реалізації задачі допускається застосування будь-якого візуального середовища програмування, але вибір його повинен бути обґрунтований.

5. Реалізація дружнього інтерфейсу є обов'язковою складовою якісно розробленої програми. Інтерфейс повинен мати: меню, елементи керування, попередження про можливі помилки при введенні інформації або під час інтерактивного режиму роботи і т.д. Всі результати вхідних, проміжних, результуючих дій повинні бути виведені на екран у вигляді, зручному для аналізу і розуміння стороннім користувачем.

6. В результаті реалізації поставленої задачі необхідно виконати аналіз роботи розробленої програми, навести висновки щодо її ефективності, умов використання, недоліків і переваг.

7. Розробка інструктивних документів для роботи з програмою є обов'язковою складовою пояснювальної записки до курсового проекту.

Результати кожного етапу виконання проекту повинні відображатися в пояснювальній записці.

## **6.2 Зміст розділу «Ескізний проект»**

Після уточнення вимог розпочинається етап проектування програмного забезпечення. Сучасні технології розробки програмних додатків базуються на структурно-орієнтованому підході при якому спочатку проводиться огляд ПрО, потім

відбувається деталізація її і приведення до деякої ієрархії об'єктів, які взаємодіють між собою за допомогою повідомлень. Через низку діаграм потоків даних (Додаток 3) визначається перелік основних процесів та їх інформаційні потоки. Деталізація закінчується на етапі коли інформаційні потоки зведені до рівня атрибутів. Визначаються сутності та типи зв'язків між ними. Далі розроблюється концептуальна модель бази даних (Додаток 4), що складається з переліку взаємопов'язаних сутностей, що використовуються для опису ПрО, разом з властивостями і характеристиками, класифікацією цих понять, за типами ситуацій, ознаками в даній області і законами перебігу процесів у ній.

Об'єкти ПрО - це конкретні предмети, а також абстрактні або реальні сутності (наприклад, замовник, замовлення, товар і т. ін.). Стан об'єкту визначає набір атрибутів, а процедури і функції – це його методи,

Кожний об'єкт є представником деякого класу однотипних об'єктів, що визначає їх загальні властивості. Усі представники (примірники) одного й того самого класу мають один і той самий набір операцій і можуть реагувати на одні й ті самі повідомлення.

Діаграма станів програми відображає доступні функції, можливі переходи програми з одного стану в інший. Кожен можливий стан коментується. Приклад діаграми станів наведено в додатку 5.

Отже в ескізному проекті подається:

дослідження потоків даних (DFD- Data Flow Diagrams);

концептуальна модель БД;

Побудова діаграми станів програми ( STD- State Transition Diagrams).

### **6.3 Зміст розділу «Технічний проект»**

Перелік робіт технічного проекту включає: розробку логічної моделі БД, вибір середовища програмування та обґрунтування його вибору, визначення переліку модулів задачі, розробку специфікації модулів, розробку архітектура програми та алгоритмів модулів.

Базою для побудови логічної моделі даних слугує концептуальна модель БД. Логічна модель є більш детальним представленням даних, в якій сутності трансформуються в таблиці даних, одна сутність може представлятися декількома таблицями, атрибути сутностей стають полями таблиць, уточнюються відношення між

таблицями, виділяються первинні ключі. Приклад опису логічної структури таблиць БД наведено в таблиці 1.

Таблиця 1 - Логічна структура таблиці «Юридичні особи»

Логічне ім'я поля	Тип	Ключ	Null/NotNull
Юридична особа	Текстовий	РК	NotNull
Адреса	Текстовий		
Керівник	Текстовий		
Головний бухгалтер	Текстовий		
Телефон керівника	Довге ціле		
Телефон бухгалтера	Текстовий		

Основними елементами інформаційної системи при використанні структурно-орієнтованого підходу є модулі. Модулі мають такі властивості:

- 1) модуль складається з множини операторів мови програмування, записаних послідовно;
- 2) модуль має ім'я, за яким до нього можна звертатися як до єдиного фрагмента;
- 3) модуль може приймати і/або передавати дані як параметри у певній послідовності чи зв'язувати дані через фіксовані осередки або загальні області.

При проектуванні системи необхідно:

розробити архітектуру програмного продукту, з урахуванням вимог щодо функцій програми, визначених інформаційних потоків та зв'язків між компонентами;

здійснити детальне проектування, яке заключається у розподілі функцій програми між модулями. Таке рішення забезпечить міцність модулів та зменшить їх зв'язаність. У разі виникнення змін у вимогах, такий розподіл забезпечить проведення доробок в програмі з мінімальними складнощами.

Для кожного модуля створюється його специфікація у складі: ім'я модуля та його призначення, вхідні дані, дія, вихідні дані, зв'язок з іншими модулями (Приклад

2)

*Приклад 2. Специфікація модулів*

Модуль «Ідентифікація користувача»

Призначення: надання санкціонованого доступу до програми та БД.

Вхідні дані: пароль.

Дія: пароль перевіряється на відповідність і при співпаданні відбувається вхід в

програму.

Вихідні дані: користувач отримує доступ до середовища вибору певної дії, або повідомлення про блокування входу у разі невірної пароля.

Зв'язок з іншими модулями: при введенні вірного пароля управління передається на модуль «Меню програми».

Алгоритми модулів можна подавати у вигляді блок-схем чи у аналітичному вигляді.

Склад технічного проекту:

ієрархічна схема модулів;

специфікація модулів;

алгоритми модулів;

логічна модель БД.

#### 6.4 Зміст розділу «Робочий проект»

На стадії робочого проекту виконується обґрунтований вибір інтегрованого середовища програмування та СКБД для реалізації бази даних. Фізична модель даних ґрунтується на логічній моделі. В ній визначається тип даних для кожного поля в таблицях даних, а також встановлюються їх властивості, таблиці нормалізуються, створюється схема БД (рис.2).

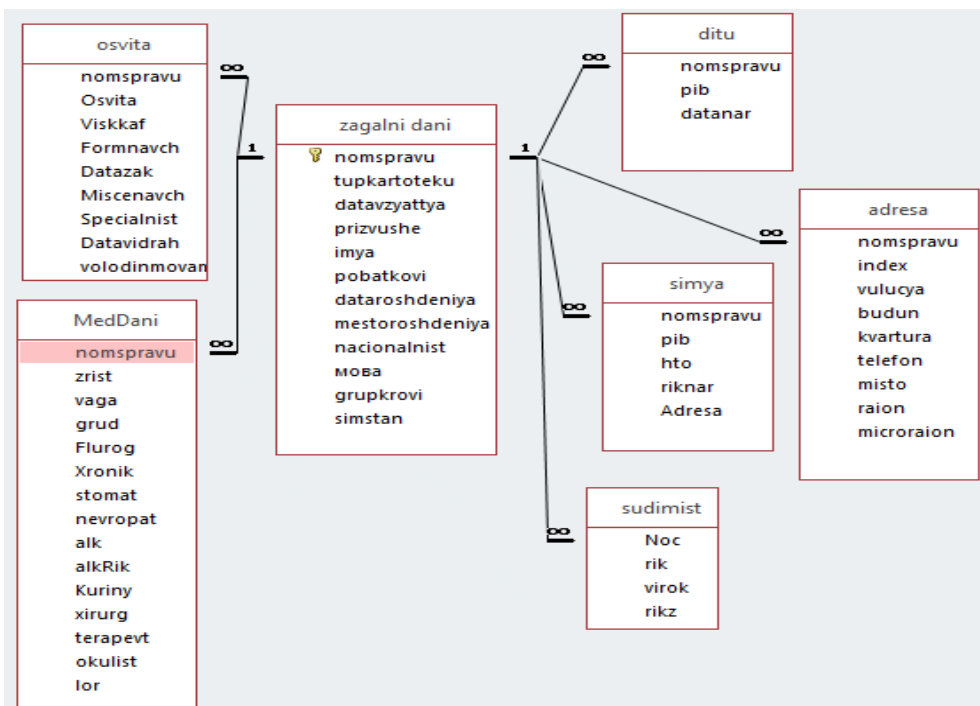


Рисунок 2 - Схема БД

Фізична структура таблиць БД подається в таблицях (Таб. 2).



Таблиця 2 - Фізична структура таблиці «Юридичні особи»

Логічне ім'я поля	Фізичне ім'я	Тип	Розмір	Ключ
Код ЮО	Kod_UO	Довге ціле	10	РК
Назва ЮО	Nazva	Текстовий	30	
Адреса	Adresa	Текстовий	30	
Керівник	Kerivnik	Текстовий	30	
...	...	...	...	...

БД наповнюється даними контрольного прикладу, який формується таким чином, щоб можна було продемонструвати формування та ведення БД, реалізацію основних запитів, звітів та виконання функцій вказаних у вимогах.

Інтерфейс користувача – це візитна картка програми, яка знаходиться на виду у користувача і покликана забезпечувати відображення даних, управління або діалог з користувачем.

Проектування інтерфейсу користувача необхідно проводити з урахуванням його високої значимості. Користувацький інтерфейс є прошарком, сполучною ланкою між людиною і програмою. Він повинен бути прозорим і зрозумілим, зручним і ефективним при реалізації завдань користувача. Час на зосвоєння інтерфейсу повинен бути мінімальним. Не треба переобтяжувати користувача зайвими попередженнями чи довідками. Необхідно враховувати, що із часом функції програми можуть бути розширені і тому інтерфейс повинен бути гнучким щодо змін.

При описі робіт по створенню програм необхідно вказати функції програми, мову і середовище програмування, системне ПЗ, та технічну базу на якому реалізовано програму.

Значну увагу необхідно приділити етапу кодування алгоритмів. Програмний код необхідно створювати з дотриманням правил написання читабельного та якісного коду (об'єктно-орієнтоване програмування, КПВ, коментування та ін.).

Опис процесу створення програми необхідно супроводжувати прикладами проектів форм, їх компонентів та фрагментами процедур обробки їх подій. Саме через обробку подій компонентів форм і реалізуються заплановані функції програми.

Кодування алгоритмів відбувається шляхом створення власних методів чи використання стандартних методів у відповідь на події об'єктів (Лістинг 1). Програмний код модулів проекту подається в додатку до пояснювальної записки.

*Лістинг 1.*

*//Процедура додавання нового запису до таблиці ЮО клік на кнопці BtAdd*

```

void __fastcall TFdov::BtAddClick(TObject *Sender)
{
int a;
a=StrToInt(Ekod->Text);
TLocateOptions top;
if (DM1->TUO->Locate("Kod_UO",a,top)==false)
{
DM1->TUO->Insert();
DM1->TUO->SetFields(ARRAYOFCONST((StrToInt(Ekod->Text),Enazv-> Text,Eadres-
>Text,Eker->Text,Ebuh->Text,StrToInt(Etker->Text),StrToInt(Etbuh-> Text))));
DM1->TUO->Post();
MessageDlg("Дані успішно несено",mtInformation,TMsgDlgButtons()<<mbOK,0);}
else
{
MessageDlg("Такий код вже існує",mtInformation,TMsgDlgButtons()<<mbOK,0);
Ekod->SetFocus();
}}

```

Опис процесів проведення обчислень, формування запитів та звітів доповнюється екранними формами, макетами запитів та звітів, Результати роботи програми на контрольному прикладі роздруковуються і подаються в додатках.

Після того, як алгоритми модулів закодовано і виконано їх налагодження на контрольному прикладі, необхідно протестувати розроблене ПЗ, адже відсутність синтаксичних помилок в програмному коді не гарантує того, що програма розроблена вірно.

Процес тестування не повинен виконуватися хаотично. Для цього існують спеціальні техніки і методики. Першим кроком вибирається метод тестування та розроблюється програма випробувань. Програму та методику випробувань ПЗ необхідно подати в додатках пояснювальної записки.

## **7 Склад програмної документації**

Завершальним розділом робочого проекту є документування програмного продукту. Програмна документація у складі «Опис програми» та «Інструкція користувачу» подається у додатках до пояснювальної записки.

Опис програми включає: загальні відомості, функціональне призначення програми, опис логічної структури програми, виклик і завантаження, вимоги до інформаційного й технічного забезпечення.

Інструкція користувача включає: вимоги до операційного середовища та технічного забезпечення, порядок інсталяції програми, послідовність виконання програми, повідомлення програми та правила їх обробки користувачем.

## **8 Порядок захисту курсового проекту**

Матеріали виконаного курсового проекту у складі: пояснювальна записка в друкованому та електронному вигляді, вихідний код програмного додатку, презентаційні матеріали студент подає на перевірку керівникові за два тижні до строку захисту. Керівник перевіряє надані матеріали на повноту, відповідність вимогам щодо оформлення та тестує програмний додаток. Демонстрація роботи розробленої студентом програми на контрольному прикладі є обов'язковою. Виявлені недоліки студент виправляє. Перевірені викладачем матеріали студент подає рецензенту за тиждень до захисту. Рецензент – це один із викладачів циклової комісії. Рецензент переглядає наведені матеріали та рецензує їх.

Захист курсового проекту проводиться відповідно до графіку, затвердженого цикловою комісією. До складу комісії входить керівник проекту та два-три викладачі циклової комісії.

На доповідь студенту дається 10-15 хвилин, протягом яких він повинен коротко обґрунтувати вибір теми, розкрити мету та основний зміст роботи, Доповідь повинна супроводжуватися презентацією результатів.

Особливу увагу в доповіді студент повинен приділити результатам розробки, її призначенню та ефекту від впровадження.

Студент повинен відповісти на практичні зауваження керівника чи рецензента (якщо вони є), погоджуючись з ними або даючи обґрунтовані заперечення. Відповіді на питання та критичні зауваження членів комісії повинні бути короткими і торкатися тільки сутності справи, відрізнятися діловитістю та науковою обґрунтованістю.

Під час виступу студент повинен продемонструвати ілюстративний матеріал, підготовлений до захисту. Бажано, щоб студент основний зміст своєї роботи викладав вільно, не читаючи письмового тексту.

## 9 Критерії оцінювання курсового проекту

Виконання курсового проекту передбачає:

1. Створення працюючого програмного продукту, який би відповідав призначенню, визначеному темою; забезпечував потрібну функціональність та мав би розвинений інтерфейс доступу користувача до означеної функціональності на рівні вимог перелічених у завданні; мав структуру, яка відповідає загальним вимогам визначених даними методичними вказівками.

2. Оформлення пояснювальної записки до проекту відповідно стандартам ДСТУ 3008-95.

3. Захист курсового проекту.

Наведений перлік покладено у основу оцінювання курсового проекту

Оцінювання курсового проекту проводиться за п'ятибальною шкалою.

Оцінку "відмінно" отримує студент, який в процесі виконання курсового проекту:

вміло, правильно і творчо застосовував теоретичні знання та практичні навички, отримані при навчанні;

використовував міжпредметні зв'язки при розв'язанні поставлених завдань;

вміло спроектував інформаційне і програмне забезпечення задачі;

виконав конструювання програмного додатку за вимогами стандартів;

закладену у вимогах функціональність програмного продукту реалізував у повному обсязі;

сформулював самостійні висновки на основі проведеного аналізу.

Студент чітко дотримувався календарного графіку виконання проекту і захистив в установлений термін.

В доповнення, така оцінка підтверджується:

1. Відповідями студента на захисті, які повинні бути:

- грамотними, чіткими і відповідати вимогам культури ділового мовлення, що свідчить про глибоке знання понятійного апарату і вміння синтезувати знання з ряду тем;

- обґрунтованим апілюванням зауважень рецензента, якщо вони є.

2. Якістю демонстраційних матеріалів та пояснювальної записки сформованою за вимогами стандартів.

Оцінка "добре" виставляється студенту, якщо курсовий проект виконаний в повному обсязі з незначними помилками, які не змінюють його суті. Зміст проекту має свідчити про вмiле і правильне застосування теоретичних знань, обґрунтоване використання міжпредметних зв'язків при його виконанні, вміння виконувати правильні розрахунки з незначними неточностями, що суттєво не впливають на результат. Таблиці, рисунки представлені з незначними порушеннями вимог щодо графічного і табличного матеріалу або на них немає посилань у тексті роботи. Сформовані висновки і рекомендації носять не завжди лаконічний і грамотний виклад результатів дослідження. Допускаються незначні відхилення від календарного графіку виконання проекту і захистом в установлений термін. Відповідь на захисті курсового проекту побудовані за вимогами і правилами культури ділового мовлення. Сприймання чи аргументоване відхилення зауважень рецензента.

Рівень знань оцінюється як "задовільно", якщо теоретичних знань недостатньо для вмiлого і творчого застосування їх при виконанні курсового проекту і для використання міжпредметних зв'язків. Підтвердженням недостатності знань є допущені помилки у розрахунках, порушення логіки і послідовності у проекті, наявність нереалізованих функцій, які сформульовані у вимогах, незавершеність висновків, незначні порушення правил культури ділового мовлення. Таблиці, рисунки представлені з незначними порушеннями вимог щодо графічного і табличного матеріалу або на них немає посилань у тексті. Сформовані висновки і рекомендації не в повній мірі відбивають суть проведеного дослідження або не містять частини отриманих результатів. Допускаються незначні відхилення від календарного графіку виконання проекту із захистом в термін, пізніший за встановлені строки. Відсутність відповіді на зауваження рецензента.

Оцінка "незадовільно" доцільна в тих ситуаціях, коли проект виконаний зі значними помилками і свідчить про незнання більшої частини матеріалу. При цьому допущені суттєві помилки в розрахунках, таблицях і викладенні змісту роботи, відсутня логіка і порушена послідовність у виконанні проекту, також висновок або відсутній, або сформульований неправильно. Проект виконаний зі значними відхиленнями від календарного графіку, при захисті студент демонструє незнання більшості матеріалу, викладеного в курсовому проекті й не вмiє роботи висновків.

## 10 Вимоги стандартів до оформлення пояснювальної записки

Курсовий проект має бути виконаний і оформлений з додержанням усіх технічних вимог. Текст роботи має бути надрукований на одному боці аркуша білого паперу формату А4 з рамкою робочого аркуша:

- *шрифт* – *Times New Roman* (для виділення прикладів, понять тощо допускається використання інших шрифтів; допускаються: напівжирний, курсив, напівжирний курсив; підкреслення не допускаються);

- *розмір шрифту* – 14;

- *відстань між рядками* – 1,5 інтервали (до 30 рядків на сторінці);

- *верхнє поле* – 15, *нижнє поле* – 30 мм, *ліве* – 25 мм, *праве* – 10 мм (не менше).

В рамці робочого аркушу записується код роботи 121.КП.П-414.08.ПЗ. Структура коду: код спеціальності-121. КП – курсовий проект, П-414 – номер групи, 08 – номер студента у списку журналу, ПЗ- пояснювальна записка.

Відстань між заголовком і текстом має бути 2 інтервали.

**Заголовки** структурних частин курсового проекту: **ЗМІСТ, ВСТУП, ВИСНОВКИ, СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** друкують великими літерами по центру до тексту.

Заголовки розділів пишуть великими літерами з абзацним відступом без крапки наприкінці. Заголовки підрозділів друкують маленькими літерами (крім першої великої) з абзацного відступу. Крапку в кінці заголовка не ставлять. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою. Назви розділів та підрозділів відокремлюють подвійним інтервалом.

Зміст пишеться на 3 сторінці письмової роботи і складається з наступних розділів: ВСТУП, ЗАГАЛЬНА ЧАСТИНА (пункти 1.1, 1.2, ...), РОЗРОБКА ПРОЕКТУ ПРОГРАМИ (пункти 2.1, 2.2, ...), ВИСНОВОК, СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ, ДОДАТКИ.

Між словом "ЗМІСТ" і самим змістом ставитися подвійний відступ. Назви розділів пишуться без відступу від лівого поля великими літерами.

Заголовки підрозділів друкують маленькими літерами (крім першої великої) з абзацного відступу. Крапку в кінці заголовка не ставлять. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою.

Відстань між заголовком та текстом повинна дорівнювати 2 інтервалам.

Абзаци в тексті починають відступом, **рівним 15 мм.**

**Нумерація** сторінок, розділів, підрозділів, пунктів, підпунктів, рисунків, таблиць подають арабськими цифрами без знака №, без крапки в кінці.

**Пункти** нумерують у межах кожного підрозділу. Номер пункту складається з номера розділу, підрозділу і пункту, між якими ставлять крапку (наприклад, 1.3.2 – другий пункт третього підрозділу першого розділу), потім у тому ж рядку йде заголовок пункту. У тексті пунктів або підпунктів можуть бути переліки. Перед кожною позицією переліку слід ставити дефіс або (за необхідності послатися в тексті на один із переліків) малу літеру, після якої ставлять дужку. Для подальшої деталізації переліку необхідно використовувати арабські цифри, після яких ставлять дужку.

Список першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня — з відступом відносно місця розташування списків першого рівня.

Приклад:

*a)*

*б)*

*1)*

*2)*

*в)*

**Графічний матеріал** — рисунки (схеми, діаграми і т. ін.) розміщують у КП для встановлення властивостей або характеристик об'єкта, а також для кращого розуміння тексту КП. На графічний матеріал мають бути посилання в тексті.

Графічний матеріал треба розмістити по центру, безпосередньо після тексту, в якому про нього згадується вперше, або на наступній сторінці, а за необхідності — у додатку.

За наявності у КП таблиць, що доповнюють графічний матеріал, таблиці слід розміщувати після графічного матеріалу.

Графічний матеріал може мати тематичну назву, яку розміщують під ним.

За необхідності, під графічним матеріалом розміщують пояснювальні дані.

Графічний матеріал слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу.

Номер рисунка складається з номерів розділу та порядкового номера рисунка, відокремлених крапкою. Наприклад: Рисунок 1.2 (другий рисунок першого розділу).

Номер ілюстрації, її назва і пояснювальні підписи розміщують послідовно під ілюстрацією. Назва пишеться після номеру, через дефіс з великої літери, без крапки наприкінці.

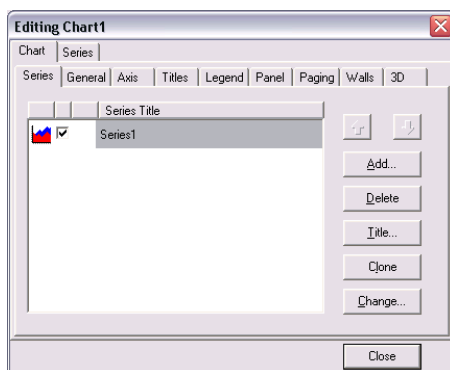


Рисунок 2.1 – Вибір типу діаграми

**Таблиці** нумерують послідовно в межах розділу. В лівому верхньому куті над відповідним заголовком таблиці розміщують напис “Таблиця” із зазначенням її номера та назви. Назва таблиці пишеться після номеру, через дефіс з великої літери, без крапки наприкінці. Номер таблиці повинен складатися з номера розділу і порядкового номера таблиці, між якими ставиться крапка, наприклад: “Таблиця 1.2” (друга таблиця першого розділу). При переносі частини таблиці на інший аркуш (сторінку) слово “Таблиця” і номер її вказують один раз над першою частиною таблиці, над іншими частинами пишуть слова “Продовження таблиці” і вказують номер таблиці, наприклад: “Продовження таблиці 1.2”.

Таблицю слід розташовувати безпосередньо після тексту, у якому вона згадується вперше, або на наступній сторінці. Назва таблиці має точно і стисло відображати її зміст.

Таблиця \_\_\_\_\_ - \_\_\_\_\_

(номер таблиці)

(назва таблиці)



На всі таблиці мають бути посилання в тексті, які складаються зі слова «таблиця» із зазначенням її номера.

Заголовки стовпців і рядків таблиці слід друкувати з великої літери,



підзаголовки стовпців — з малої, якщо вони є продовженням заголовка, або з великої, якщо вони мають самостійне значення. У кінці заголовків і підзаголовків таблиць крапки не ставлять, заголовки і підзаголовки стовпців друкують в однині.

Таблиці ліворуч, праворуч і знизу, як правило, обмежують лініями.

Розділяти заголовки і підзаголовки стовпців діагональними лініями не допускається.

Заголовки стовпців, як правило, друкують паралельно рядкам таблиці. За необхідності допускається перпендикулярне розміщення заголовків стовпців.

Допускається розміщення таблиці вздовж довгого боку аркуша.

Якщо рядки або стовпці таблиці виходять за формат сторінки, то таблицю ділять на частини, які розміщують одна під одною або поряд, при цьому в кожній частині таблиці повторюють її заголовок.

У разі поділу таблиці на частини її заголовки заміняють відповідно номерами стовпців. При цьому нумерують арабськими цифрами стовпці першої частини таблиці.

**Формули** в курсовому проекті (якщо їх більше одної) нумерують у межах розділу. Номер формули складається з номера розділу і порядкового номера формули в розділі, між якими ставлять крапку. Нумери формул пишуть біля правого поля аркуша на рівні відповідної формули в круглих дужках, наприклад: (3.1) (перша формула третього розділу).

Посилання в тексті на порядкові номери формули дають у дужках.

*Приклад:*

...у формулі (1.1).

У формулі як символи фізичних величин слід застосовувати позначення, встановлені відповідними стандартами або іншими документами.

Для економії місця кілька коротких однотипних формул, відокремлених від тексту, можна подати в одному рядку, а не одну під одною. Невеликі і нескладні формули, що не мають самостійного значення, вписують всередині рядків тексту. Пояснення значень символів і числових коефіцієнтів треба подавати безпосередньо під формулою в тій послідовності, в якій вони дані у формулі. Значення кожного символу і числового коефіцієнту треба подавати з нового рядка. Перший рядок пояснення починають без абзацу зі слова “де” без двокрапки.

Рівняння і формули треба виділяти з тексту вільними рядками. Вище і нижче

кожної формули потрібно залишити не менше одного вільного рядка.

Формули, що подаються одна за одною і не розділені текстом, відокремлюють комою.

Наприклад:

$$N = \frac{T}{F_{ef} \cdot K_{en}}, \quad (1.1)$$

де  $N$  – необхідна чисельність розробників ПЗ, чол.;

$T$  – загальна трудомісткість, норм-год;

$F_{ef}$  – ефективність фонду робочого часу одного працюючого;

$K_{en}$  – коефіцієнт виконання норм.

## 11 Список літературних джерел

1. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: Держстандарт України, 1995. – 38 с..
2. Міжнародний стандарт ISO 12207.
3. ДСТУ 2844-94 Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення;
4. ДСТУ 2873-94 Системи оброблення інформації. Програмування. Терміни та визначення;
5. Шквір В.Д. та ін. Інформаційні системи і технології в обліку. Львів, НУ, 2003, 268 с..
6. Хомоненко А.Д. і др. Базы данных. С-П, Корона принт, 2002, 662.3.стр.
7. Грег Перри. Самоучитель программирования. Питер, 2002 362 стр.
8. Орлов С.А. Технологии разработки программного обеспечения: Учебник/СПб: Питер, 2002. 464 с. ил.
9. Зингер К. Методы проектирования программных систем. М. Мир, 1982. 3–328 с.
10. Калянов Г.Н. CASE структурный системный анализ. М. ЛОРИ, 1996.–242 с
11. Андон Ф.И., Лаврищева Е.М. Методы инженерии распределенных компьютерных приложений. – Киев: Наук. думка, 1998. – 228 с.
12. Бабенко Л.П., Лаврищева К.М. Основы программной инженерии. Посібник.– К.: Знання, 2001.– 269с.
13. Соммервил И. Инженерия программного обеспечения.– Изд. дом „Вильямс”, Москва+ Санкт–Петербург+ Киев. – 2002. – 623 с.

14. Введение в программную инженерию и управление жизненным циклом программного обеспечения (SWEBOOK): Пер. с англ. С.Орлик [Электронный ресурс]. – [sorlik.blogspot.com/](http://sorlik.blogspot.com/)

## **Додаток 1 Орієнтований перелік тем курсових проектів**

1. Розробка комплексу програм з використанням основних принципів об'єктно-орієнтованого програмування та ІЗВП для роботи з графічними об'єктами
2. Розробка ПЗ формування та підтримки в актуальному стані електронного освітнього ресурсу.
3. Розробка комплексу програм автоматизації операцій обліку замовлень в торгово-закупівельній фірмі.
4. Розробка програмного додатку до БД «Призовник».
5. Розробка програмного забезпечення задачі «Облік основних засобів виробництва».
6. Розробка програмного забезпечення АРМ-психолога навчального закладу
7. Розробка ПЗ задачі «Облік пільговиків в сільраді»
8. Розробка ІПС задачі «Автоматизація обліку продажу квитків в касах автовокзалу»
9. Розробка комплексу програм для автоматизації операцій обліку робочого часу та нарахування заробітної плати держслужбовцям.
10. Розробка комплексу програм обліку роботи обладнання в цеху.
11. Розробка програмного додатку до БД «Автопідприємство».
12. Розробка програмного додатку до БД «Хлібозавод»
13. Розробка ІПС «Абітурієнт»
14. Розробка програмного забезпечення ІПС «Кадри»
15. Розробка програмного забезпечення задачі «Автоматизація обліку успішності студентів»
16. Автоматизація операцій обробки подорожніх листів.
17. Розрахунок потреби в матеріалах та комплектуючих під план випуску продукції;
18. Розробка алгоритмічного та програмного забезпечення задачі «Розробка графіка проведення технологічного огляду та ремонту обладнання»
19. Розробка алгоритмічного та програмного забезпечення задачі «Обробка показників датчиків контролю роботи технологічного обладнання»;
20. Розробка алгоритмічного та програмного забезпечення ІПС «Поліклініка»

## Додаток 2 Приклад завдання на курсовий проект

Державний вищий навчальний заклад

**«Миколаївський політехнічний коледж»**

(повне найменування вищого навчального закладу)

Відділення Економічне

Циклова комісія Програмування

Освітньо-кваліфікаційний рівень Молодший спеціаліст

Напрямок підготовки 050103 «Програмна інженерія»

Спеціальність 5.05010301 «Розробка програмного забезпечення»

### ЗАВДАННЯ

на курсовий проект (роботу) студенту

**Іванченко Дмитра Степановича**

(прізвище, ім'я, по батькові)

1. Тема проекту Розробка програмного забезпечення задач АРМ-інспектора ФСС

Керівник проекту: Ст. викладач вищої категорії Бойченко Сергій Іванович

(прізвище, ім'я, по батькові науковий ступень, вчене звання)

затверджені наказом вищого навчального закладу від «\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Строк подання студентом проекту \_\_\_\_\_

3. Вихідні дані до проекту: Розробити програмне забезпечення задачі «Контроль використання коштів виділених ФСС за заявами юридичних осіб» . Передбачити функції формування та ведення БД, формування запитів та звітів за критеріями користувача. Протестувати розробку на контрольному прикладі. Тексти програмних модулів, програму та методику тестування подати в додатках

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Вступ

Розділ 1. Загальна частина

1.1 Опис предметної області

1.2 Формулювання вимог до програми

1.3 Технічне завдання

Розділ 2. Розробка проекту програми

2.1 Ескізний проект

2.2 Технічний проект

2.3 Робочий проект

2.4 Результати роботи програми

2.5 Опис програми

2.6 Інструкція користувачу

2.7 Список використаної літератури

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Виготовлені з використанням Case-засобів діаграми: варіантів використання, потоків даних, станів програми, архітектура програми, схема взаємодії модулів програми, концептуальна модель БД.

2. Презентаційні матеріали

Розглянуто і схвалено цикловою комісією Програмування

Протокол № \_\_\_\_\_ від «\_\_» \_\_\_\_\_ 20\_\_ р.

Студент \_\_\_\_\_ (підпис)

Керівник проекту \_\_\_\_\_ (підпис)

### Додаток 3. Приклад діаграм потоків даних

Модель потоків даних – представляє процес у вигляді набору дій, кожна з яких виконує деяке перетворення даних. Діаграми потоків даних (DFD) є основним засобом моделювання функціональних вимог майбутнього ПЗ. З їх допомогою ці вимоги розбиваються на функціональні компоненти (процеси) і представляються у вигляді мережі, зв'язаної потоками даних.

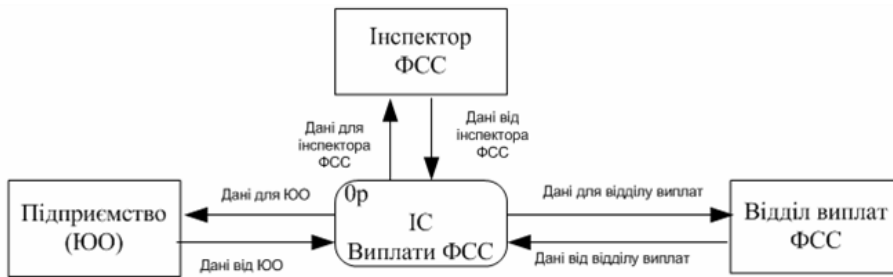


Рисунок Д3.1 - Діаграма 0 – го рівня. Контекстна діаграма

Зміст інформаційних потоків діаграми 0-го рівня наступний:

Дані від юридичної особи: заяви на нарахування коштів ФСС, звіти про використання коштів.

Дані для юридичної особи: платіжні доручення на перерахування коштів, акти перевірки використання коштів та нарахування штрафів.

Дані від інспектора інформація за результатами перевірки, обсяги нарахування коштів для ЮО за заявками, результати перевірки використання коштів.

Дані для інспектора інформація ЮО про використання коштів ФСС, напрями і обсяги використаних фондів, результати розрахунків.

Дані для відділу нарахувань: інформація про обсяги замовлених коштів за напрямами по юридичним особам, результати запитів, звіти.

Дані від відділу нарахувань: інформація для платіжних доручень по напрямам та для юридичних осіб.

На рисунку Д3.2 приведена діаграма першого рівня, яка дає основне уявлення про функціонування системи.

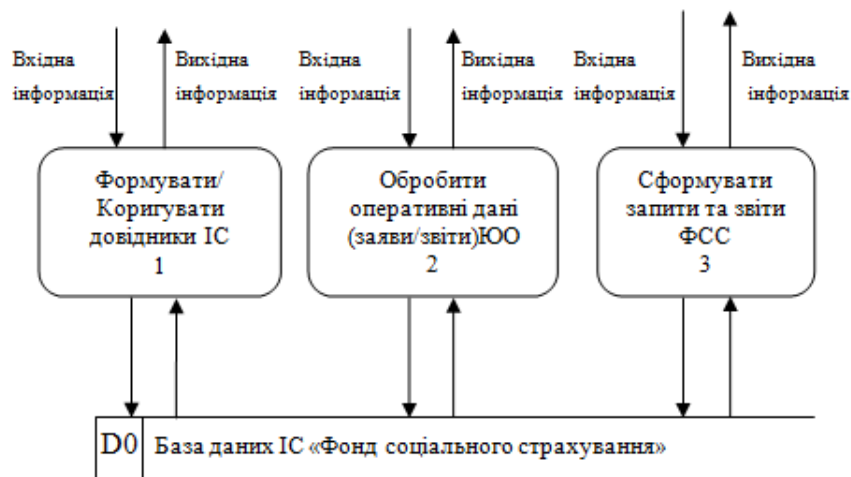


Рисунок Д3.2 - Діаграма першого рівня основного процесу

Вхідні інформаційні потоки несуть у собі дані від юридичних осіб, відділу виплат та інспектора ФСС. Вихідні інформаційні потоки являють собою результати роботи програми.

Процес «Формувати (коригувати) довідники» має підпроцеси по роботі з довідником юридичних осіб та довідником напрямів виплат. Діаграма 2-го рівня цього процесу наведена на (рисунок Д3.3).



Рисунок ДЗ.3 - Діаграма 2 –го рівня процесу «Формувати (коригувати) довідники»

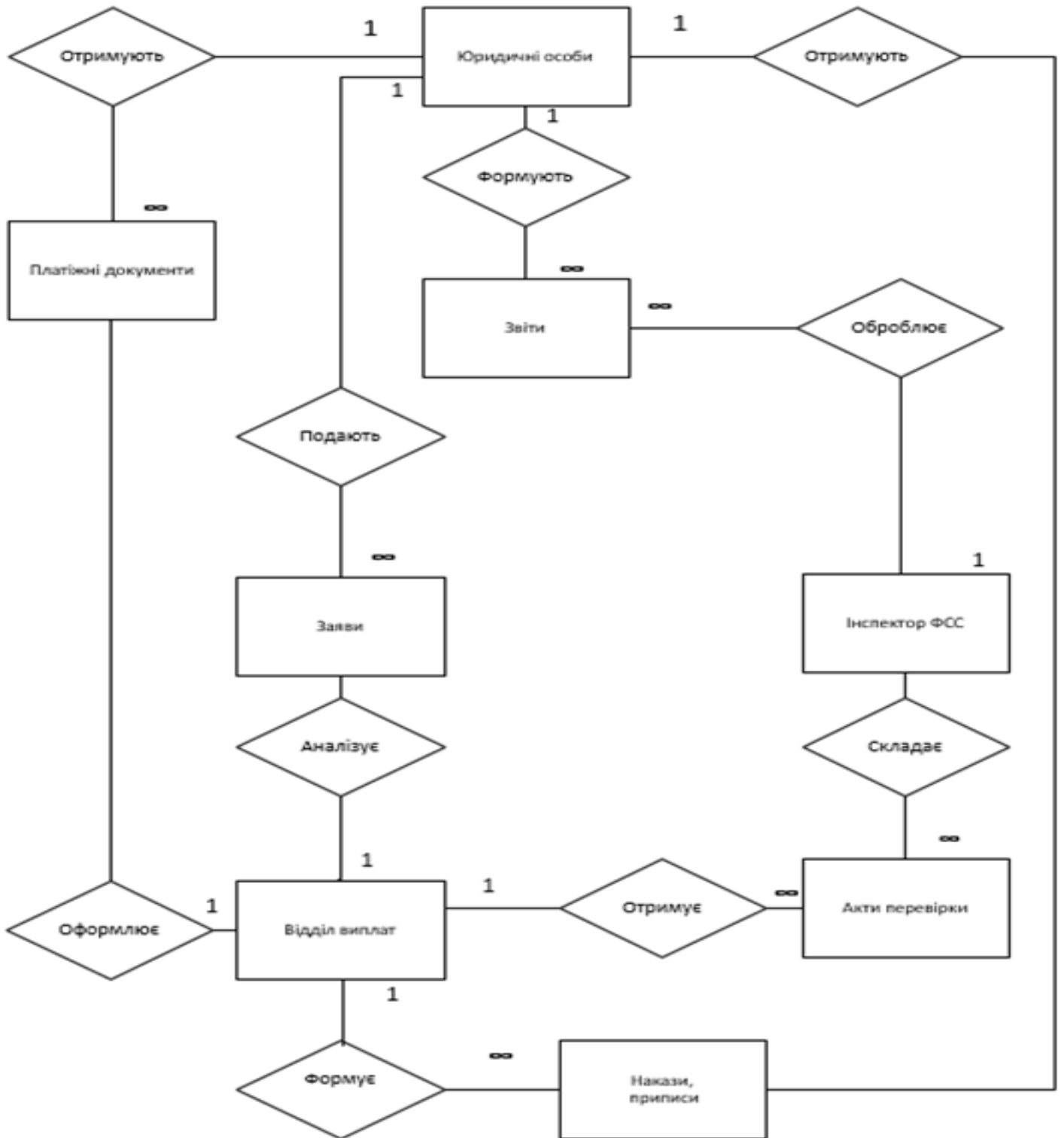
Зміст вхідної і вихідної інформації процесу наведено в словнику

Таблиця 1 – Словник DFD 2-го рівня задачі формування та ведення довідників БД

Потоки даних ДПД 1 рівня	Потоки даних ДПД 2 рівня	Атрибути
1	2	3
Вхідна інформація	Дані ЮО	Код юридичної особи, назва ЮО, адреса. керівник, гол. бухгалтер телефон керівника, телефон бухгалтера.
	Перелік витрат	Код витрат, напрям витрат.
Вихідна інформація	Запис в таблицю «Юридичні особи»	Код юридичної особи, назва ЮО, адреса керівник, гол. бухгалтер телефон керівника, телефон бухгалтера.
	Табл. «Напрями витрат»	Код витрат, напрям витрат.

Процес приведено до «простих» операцій і в подальша деталізація не потрібна.

# Додаток 4 Приклад концептуальної моделі БД





## Додаток 5 Приклад діаграми станів програми

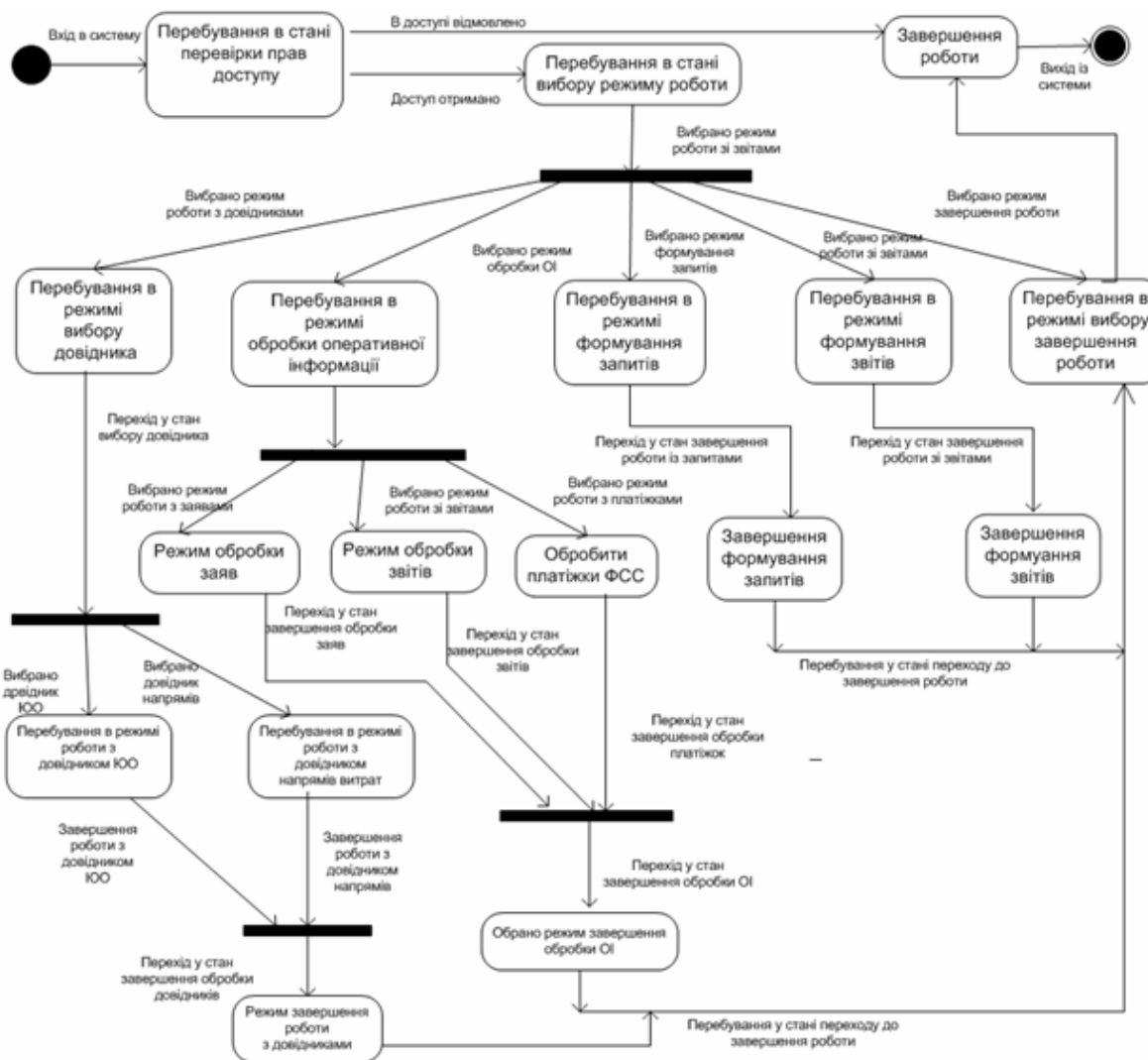


Рисунок Д.5 - Діаграма станів програми

Початковий стан програми – перевірка прав доступу в систему. Якщо користувач не знає паролю для входу в програму то йому відмовляється в доступі і програма переходить у стан завершення роботи програми.

Стан надання режимів роботи забезпечує користувача інструментами переходу у вибраний режим роботи. По завершенню роботи програми у вибраному режимі відбувається вихід на меню програми.

Режими роботи програми наступні: робота з довідниками, формування запитів, контроль ресурсів ФСС, перегляд звітів.

## Додаток 6. Приклад розділу «Програма та методика тестування»

### 1.Об'єкт випробувань

Об'єктом випробувань є програмне забезпечення інформаційно – пошукової системи «Каси Автовокзалу».

### 2.Мета та ціль випробувань

Програма випробувань програмного забезпечення переслідує наступні цілі:

перевірка придатності розробленого програмного забезпечення до використання;

перевірка придатності розробленого програмного забезпечення для рішення задач у предметній області, для якої даний програмний продукт був створений;

перевірка досягнутих характеристик програмного продукту.

Метою випробувань є оцінка відповідності програмного забезпечення вимогам, сформульованим у технічному завданні, наведеному у відповідному розділі, і виявлення помилок проектування. Якщо прийнятий рівень відповідності програмного забезпечення досягнутий, а також відсутні помилки і грубі недоліки, то програмне забезпечення приймається в експлуатацію.

### 3. Склад документів, що надаються для проведення тестування

Для проведення випробувань надаються наступні документи:

- технічне завдання на створення програмного забезпечення;
- проект програмного забезпечення;
- початкові тести програмного забезпечення;
- керівництво користувача.

### 4. Технічні вимоги

Програмні модулі формування та ведення довідників, документів, звітів та оброблення даних повинні передбачати функціонування програмного забезпечення в діалоговому режимі. Тривалі процедури повинні супроводжуватися діагностичними повідомленнями. Видавати повідомлення і попередження у критичних ситуаціях.

### 5. Методи випробувань

Випробування проводяться після завершення етапу тестування і налагодження програмного забезпечення розробником з групою користувачів автовокзалу, працівників кас автовокзалу.

Програма тестування створена на основі визначеного набору простих правил, які дозволяють одержати множину перевірочних тестів. Ці правила базуються на розгляді програмних модулів як «чорної скрині» і використанні їх зовнішніх специфікацій для побудови тестів. На базі вивчення програми модуля будуються додаткові тести. Тести створювалися методом розбиття на класи еквівалентності (грунтується на припущенні, що якщо виконується тест для одного зі значень класу еквівалентності, то він буде аналогічно виконуватися й для всього класу) та припущення про помилки (грунтується на інтуїції й досвіді типових помилок).

Під час тестування визначались вхідні дані, їх перетворення програмою та форма запису до БД вихідних даних. На основі визначалася працездатність модулів.

Приведемо тест модулю меню "Menu". У таблиці – 1 представлені виділені класи еквівалентності.

Таблиця 1 – Виділені класи еквівалентності

	Вхідні дані	Вірні класи	Невірні класи
1	Пароль	>0	
2		Введений пароль цифрами	
3			=0
4			Введений пароль літерами літерами

Результати тестування на основі виділених класів еквівалентності подано у таблиці – 2.

Таблиця 2.– Тести з вірних та невірних класів еквівалентності

№ теста	Класи	Вхідні дані	Вихідні дані
1	1	1111	Вхід до меню
2	2	0000	Повідомлення. Не вірний пароль
3	3	11ук	Повідомлення. Не вірний пароль

Тестування модуля «Каса», формування таблиці «Продаж». У таблиці – 3 представлені виділені класи еквівалентності.

Таблиця 3 - Виділені класи еквівалентності модуля – «Каса»

Вхідні дані	Вірні класи	Невірні класи
Н.Рейсу	>0	0, не задано
Д.Продажу	Дата	“ “, не задано
Д.Рейсу	Дата	Не задано
Н.Місця	Цілочислове значення	Задано літеру в кодї
Н.Каси	Цілочислове значення	Не задано
Ціна	Цілочислове значення	Не задано
П.Відправлення	Текст	Не задано
П.Призначення	Текст	Не задано

Результат тестування на основі виділених класів еквівалентності приведені у таблиці – 4-5.

Таблиця 4 – Тест з невірним класом еквівалентності

Класи	Вхідні дані	Вихідні дані
Н.Рейсу	34	Повідомлення. Не відповідність полів
Д.Продажу	20	
Д.Рейсу	Не введено	
Н.Місця	Не введено	
Н.Каси	5	
Ціна	«»	
П.Відправлення	«»	
П.Призначення	«»	

У таблиці – 5 наведений тест з вірним класом еквівалентності.

Таблиця 5 – Тест з вірним класом еквівалентності

Класи	Вхідні дані	Вихідні дані
Н.Рейсу	345	Дані запишуться
Д.Продажу	04.04.2013	
Д.Рейсу	04.04.2013	
Н.Місця	6	
Н.Каси	2	
Ціна	23	
П.Відправлення	Миколаїв	
П.Призначення	Баштанка	

Протестуємо документ модуля «Каса», таблиці «Квиток». У таблиці – 6 представлені виділені класи еквівалентності.

Таблиця 6 – Виділені класи еквівалентності модуля – «Каса»

Вхідні дані	Вірні класи	Невірні класи
Н.Каси	4	0, не задано
П.Відправлення	Текст	“ “, не задано
П.Призначення	Текст	Не задано
Відправлення	Дата	Задано літеру в кодї
Н.Місця	Цілочислове значення	Не задано
Вартість	Цілочислове значення	Не задано

Протестуємо модуль «Довідники» таблицю «Автобуси».

Таблиця 7 – Виділені класи еквівалентності таблиці «Автобуси».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Державний номер	Цифри > 0	=0, введені літери
2	Код моделі	Цифри > 0	=0, введені літери
3	Власник	Текст	Введені цифри

Таблиця 8 – Виділені класи еквівалентності таблиці «Каса».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Номер каси	Цифри > 0	=0, введені літери
2	Касир	Текст	введені цифри

Таблиця 9 – Виділені класи еквівалентності таблиці «Маршрут».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Н.Рейсу	Цифри > 0	=0, введені літери
2	П.А	Текст	Введені цифри, пусто
3	Прибув	Дата	Введені цифри, пусто, введені літери
4	Стоянка	Цифри	Введені літери, пусто
5	Відправлення	Дата	Введені цифри, пусто, введені літери
6	В дорозі	Цифри	Введені літери, пусто
7	Вартість	Цифри	Введені літери, пусто

Таблиця 10 – Виділені класи еквівалентності таблиці «Моделі».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Код моделі	Цифри > 0	=0, введені літери
2	Модель	Текст, цифри	пусто
3	Рядів	цифри	Введені літери, пусто
4	Місць	Цифри	Введені літери, пусто

Таблиця 11 – Виділені класи еквівалентності таблиці «Продаж».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Н.Рейсу	Цифри > 0	=0, введені літери
2	Дата П.	Дата	введені цифри, літери, пусто
3	Дата Р.	Дата	введені цифри, літери, пусто
4	Н.Місця	число	введені літери, пусто
5	Н.Каси	число	введені літери, пусто
6	Ціна	число	введені літери, пусто
7	П.Відправлення	Текст	введені цифри, пусто
8	П.Призначення	Текст	введені цифри, пусто

Таблиця 12 – Виділені класи еквівалентності таблиці «Рейси».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Н.Рейсу	Цифри > 0	=0, введені літери
2	П.Старту	Текст	введені цифри, пусто
3	П.Призначення	Текст	введені цифри, пусто
4	Час старту	Дата	введені літери, цифри пусто
5	Час фінішу	Дата	введені літери, цифри пусто

Таблиця 13 – Виділені класи еквівалентності таблиці «Розподіл».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Державний номер	Цифри, числа	=0
2	Н.Рейсу	Число	введені літери, пусто
3	Дата	Дата	введені цифри, літери, пусто
4	Водій 1	Текст	введені цифри, пусто
5	Водій 2	Текст	введені цифри, пусто

Таблиця 14 – Виділені класи еквівалентності таблиці «Водії».

№ п/п	Вхідні дані	Вірні класи	Невірні класи
1	Таб.Номер	Цифри > 0(...)	=0, введені літери
2	ПІБ	Текст	введені цифри, пусто
3	Клас	Цифри	введені літери, пусто
4	Права	Цифри	введені літери, пусто
5	Адреса	Текст	введені цифри, пусто
6	Дата Народження	Дата	введені літери, пусто
7	Телефон	Цифри(.._.._..)	введені літери, пусто