

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЗ „ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА”**

Кафедра інформаційних технологій та систем  
(назва кафедри)

Програмування та ООП  
(назва дисципліни)

для освітнього рівня бакалавр

напряму / спеціальності(ей) 123 – Комп’ютерна інженерія

**ЗАТВЕРДЖЕНО**

на засіданні кафедри  
Протокол № 1 від 30.08.19

Завідувач кафедри \_\_\_\_\_ (підпис)

Перезатверджено: протокол № \_\_\_\_\_ від \_\_\_\_\_

Перезатверджено: протокол № \_\_\_\_\_ від \_\_\_\_\_

Перезатверджено: протокол № \_\_\_\_\_ від \_\_\_\_\_

**Старобільськ – 2019**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ ЗАКЛАД  
ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Методичні вказівки**

до виконання курсової роботи з дисципліни

«Програмування та ООП»

для студентів спеціальності

«123 Комп'ютерна інженерія»

**Старобільськ  
«Альма-матер»  
2019**

УДК [004:519.6](076)  
ББК 22.19 р3:32.973 р3  
К 77

**Рецензенти:**

*Могильний Г. А.* — кандидат технічних наук, доцент, завідувач кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка.

*Ляхно В. А.* — кандидат технічних наук, доцент кафедри економічної кібернетики Луганського національного аграрного університету.

**К 77 Донченко В.Ю. Курсовий проект:** Методичні рекомендації до виконання курсового проекту з об'єктно-орієнтованого програмування для студентів спеціальності 123 Комп'ютерна інженерія. — Старобільськ: Альма-матер, 2018. — с.

Методичні рекомендації призначені для виконання курсового проекту з об'єктно-орієнтованого програмування. Методичні рекомендації містять орієнтовну тематику курсових проектів, організацію виконання, структуру пояснювальної записки, змісти розділів, вимоги згідно ЄСКД України до оформлення пояснювальної записки курсового проекту, приклади оформлення окремих структурних елементів. Мета методичних рекомендацій — допомогти студентам при виконанні курсової роботи, призначеної для закріплення знань, отриманих при вивченні теоретичної частини курсу, і отримання практичних навичок розробки об'єктно-орієнтованих програм, а також оформленні пояснювальної записки.

Рекомендовані для студентів спеціальності 123 Комп'ютерна інженерія, 6.050103 «Програмна інженерія», можуть бути використані студентами інших спеціальностей.

УДК [004:519.6](076)  
ББК 22.19 р3:32.973 р3

*Рекомендовано до друку навчально-методичною радою  
Луганського національного університету  
імені Тараса Шевченка*

© Донченко В.Ю.

© Альма-матер, 2019

## ЗМІСТ

ВСТУП.....	5
МЕТА ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ.....	5
1. ЗАВДАННЯ НА ПРОЕКТУВАННЯ.....	7
2. ПРИКЛАД ВИКОНАННЯ ПРОЕКТУ .....	8
2.1. Постановка завдання.....	8
2.2. Опис предметної області.....	9
2.3. Побудова об'єктної моделі.....	10
2.4. Визначення об'єктів і класів .....	10
2.5. Підготовка словника даних.....	11
2.6. Визначення залежностей (зв'язків) між класами .....	12
2.7. Побудова діаграми ієрархії класів.....	14
3. ВАРІАНТИ ЗАВДАНЬ НА КУРСОВУ РОБОТУ.....	15
4.ВИМОГИ ДО ЗМІСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ .....	26
5. ЛІТЕРАТУРА .....	32

## **ВСТУП**

Курсова робота - це перша самостійна наукова праця майбутнього програмного інженера. При її виконанні студент поглиблює знання з фундаментальних та професійно-орієнтованих дисциплін, засвоює методику експериментальних досліджень, оволодіває навичками співставлення результатів своїх досліджень із літературними даними, аналізу, узагальнення і літературного оформлення одержаних результатів з теми дослідження, набуває вміння вести науковий пошук, що розвиває в молодого інженера творчий підхід до роботи.

Курсова робота з об'єктно-орієнтованого програмування передбачає знання студентами об'єктно-орієнтованого проектування та програмування і є завершальним етапом вивчення у рамках курсу об'єктно -орієнтоване програмування підготовки студентів спеціальності 123 Комп'ютерна інженерія, 6.050103 «Програмна інженерія» кваліфікаційного рівня „Бакалавр” та виконується протягом III семестру.

### **МЕТА ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ**

Метою курсової роботи є закріплення, поглиблення та узагальнення знань, якими студент оволодів під час вивчення курсу, в набутті навичок використання основ алгоритмізації та програмування на алгоритмічних мовах високого рівня з використанням принципів об'єктно-орієнтованого проектування та програмування, методики розробки програмних модулів та програмного забезпечення комп'ютерних систем управління, набуття студентами теоретичних знань та практичних навичок в області використання сучасних систем візуального проектування програмних засобів комп'ютерних систем управління та освоєння принципів та методів сучасних технологій програмування, набуття навичок професійних та практики наукових досліджень з оформленням відповідного текстового, програмного та ілюстративного матеріалу у формі наукового звіту.

Виконання курсової роботи (КР) направлено на закріплення знань, отриманих при вивченні теоретичної частини курсу, і набуття практичних

навичок розробки об'єктно-орієнтованих програм. В результаті виконання курсової роботи студент повинен освоїти:

- концепції об'єктно-орієнтованого програмування;
- спеціалізовані засоби розробки програмного забезпечення мовою C +;
- основи уніфікованої мови моделювання UML.

**знати:** загальні питання побудови та використання сучасних систем об'єктно-орієнтованого та візуального проектування програмних засобів комп'ютерних систем управління; основні поняття та методи об'єктно-орієнтованого проектування та програмування, основні засоби розробки програм з використанням багато ієрархічних структур об'єктів та класів об'єктів;

**вміти:** самостійно вибирати середовище проектування програмних засобів, проектувати, розробляти, налагоджувати та супроводжувати сучасне програмне забезпечення комп'ютерних систем управління.

**Під час виконання** курсової роботи студент повинен продемонструвати:

- вміння збирати і аналізувати відповідні матеріали про об'єкт дослідження, використовуючи сучасні джерела інформації, включаючи Інтернет ресурси;
- спроможність проводити необхідні обґрунтування для розробки програмних засобів систем управління різного призначення, тощо;
- здатність доводити розв'язання поставленої задачі до логічного кінця;
- - вміння аналізувати отримані результати і робити з них висновки.

КР складається з двох взаємопов'язаних частин. Для виконання першої частини КР потрібно побудувати детальну об'єктну модель запропонованої предметної області. Друга частина КР передбачає безпосередню розробку

об'єктно- орієнтованого програмного забезпечення, реалізує розроблену об'єктну модель.

## **1. ЗАВДАННЯ НА ПРОЕКТУВАННЯ**

Варіант завдання на КР студенту призначає викладач. Постановки завдань наведені в розділі 3.

Кожна з двох частин КР розглядається, як самостійне завдання і може захищатися окремо. Перша частина КР вирішує питання аналізу та проектування системи, моделює реальну предметну область. Так як запропонований варіант формулювання задачі являє собою лише поверхневий опис предметної області і не претендує на повноту і завершеність, то конкретизація і уточнення покладаються на проектувальника. При цьому може виявитися, що функціональна повнота (забезпечення всіх необхідних характеристик) не може бути досягнута за обмежений час, виділений на розробку. У цьому випадку необхідно узгодити з викладачем підмножину функцій, які будуть складати ядро проекрованої системи.

У першій частині КР необхідно:

- зробити опис запропонованої в завданні предметної області;
- визначити набір функцій, які повинна вирішувати система описуваної предметної області;
- виділити з опису предметної області можливі об'єкти і класи;
- підготувати словник даних;
- визначити залежності (зв'язку) між об'єктами (класами);
- визначити атрибути об'єктів;
- виконати групування і запропонувати ієрархію спадкування класів;
- провести подальше дослідження та удосконалення отриманої моделі;
- написання «скелета» всіх отриманих класів.

Друга частина - безпосередня розробка програмного забезпечення базується на результатах проектування виконаного в першій частині.

При виконанні курсового проекту рекомендується спиратися на бібліографічні джерела [1-5] та іншу літературу з об'єктно-орієнтованого програмування.

Незалежно від варіанту завдання на КР можна виділити кілька формальних критеріїв оцінки програми розробленої студентом:

**Коректність програми.**

Програма правильно працює з коректними тестовими даними.

**Стійкість програми.**

Введена інформація піддається логічному контролю. Програма не повинна втрачати працездатності, при будь-яких діях, навіть некоректних діях користувача. Всі дії, які загрожують втратою інформації, виконуються тільки після повторного підтвердження.

**Функціональна повнота.**

Програма повинна здійснювати протоколювання всіх виконуваних в ній дій. В рамках погодженою з викладачем підмножиною функцій, всі вони повинні бути реалізовані (Приклад: програма повинна дозволяти проводити пошук або вибірку інформації по зробленому фільтру. При невдалому завданні умови запиту забезпечується його коригування, з метою звуження або розширення зони пошуку).

**Засоби допомоги і документації.**

Програма повинна бути забезпечена внутрішньою документацією у вигляді коментарів. При запуску програми повинна з'являтися заставка, що відображає суть і можливості програмного засобу, а також відомості про автора. Бажано передбачити можливість існування контекстно-залежної допомоги.

## **2. ПРИКЛАД ВИКОНАННЯ ПРОЕКТУ**

### **2.1. Постановка завдання**

Банк. Відомості про вкладників банку: номер особового рахунку, категорія вкладу, паспортні дані, поточна сума внеску, дата останньої



операції. Операції прийому і видачі будь-якої суми, автоматичне нарахування відсотків.

## **2.2. Опис предметної області**

Клієнти банку мають пластикові банківські картки (один клієнт може мати декілька карток); картка містить код картки, код банку, код клієнта та іншу інформацію, що забезпечує доступ до рахунку (рахунках) клієнта в цьому банку. Клієнт може вкласти свою картку в термінал (банкомат) і, за умови, що код картки і код банку вірні, почати банківську проводку.

Проводка полягає в узгодженій зміні даних на рахунках клієнта та звітної документації банку, що зберігаються в базі даних банку, відповідно за даними проводки. Проводка включає в себе і перевірку права клієнта на доступ до його рахунках на момент проведення (перевірка безпеки), і перевірку відповідності суми, затребуваної клієнтом, поточним станом його рахунку. Якщо перевірки пройшли успішно, клієнт отримує в терміналі затребувану ним суму грошей і квитанцію, в іншому випадку він отримує тільки квитанцію. Під час здійснення проводки можуть відбутися збої в роботі апаратури, або клієнт може передумати отримувати гроші і скасувати вже проводку, яка почалася. У цьому випадку всі рахунки і звітні документи повинні бути відновлені в тому стані, в якому вони були до початку проведення (відкат).

Для реалізації відкату використовується служба ведення записів про зміни, внесених до бази даних банку при виконанні проводки. Всі дії, пов'язані з виконанням

проводки (в тому числі протоколювання і забезпечення безпеки проводки), виробляються програмним забезпеченням системи управління банківською мережею.

Комп'ютер банку підтримує рахунки клієнтів, тобто зберігає їх у своїй базі даних і виконує проводки над цими рахунками за запитами з банкомату (Дистанційна проводка) або з касових терміналів (проведення касира, дані про які вводяться касиром).

### **2.3. Побудова об'єктної моделі**

Розглянемо процес побудови об'єктної моделі для системи банківського обслуговування в процесі аналізу вимог і попереднього проектування цієї системи.

Для побудови об'єктної моделі даної системи необхідно виконати наступні етапи:

- визначення об'єктів і класів;
- підготовка словника даних;
- визначення залежностей між об'єктами;
- визначення атрибутів об'єктів і зв'язків;
- організація та спрощення класів при використанні спадкування;
- подальше дослідження і вдосконалення моделі.

### **2.4. Визначення об'єктів і класів**

Аналізуючи постановку завдання і предметну область, можна виділити набір можливих класів:

термінал, (банкомат) касир  
програмне забезпечення  
банк, касовий термінал, система  
банківська мережа, квитанція, перевірка безпеки  
дані проводки, клієнт, служба ведення записів  
дані, рахунки, комп'ютер банку, рахунок  
гроші, центральний комп'ютер  
ціна, доступ  
користувач, картка, проводка

З цього набору слід виключити надлишкові класи, нерелевантні класи, нечітко визначені класи, можливі атрибути, реалізаційні конструкції. Після виключення всіх зайвих імен можливих класів, отримуємо уточнений список класів, що становлять проектувану систему банківського обслуговування:

термінал (банкомат), касовий термінал  
проводка, банк, клієнт

рахунок, картка, комп'ютер банку, касир

Надалі цей набір класів може бути доповнений або змінений. При використанні візуального середовища проектування необхідно в ній додати всі запропоновані класи.

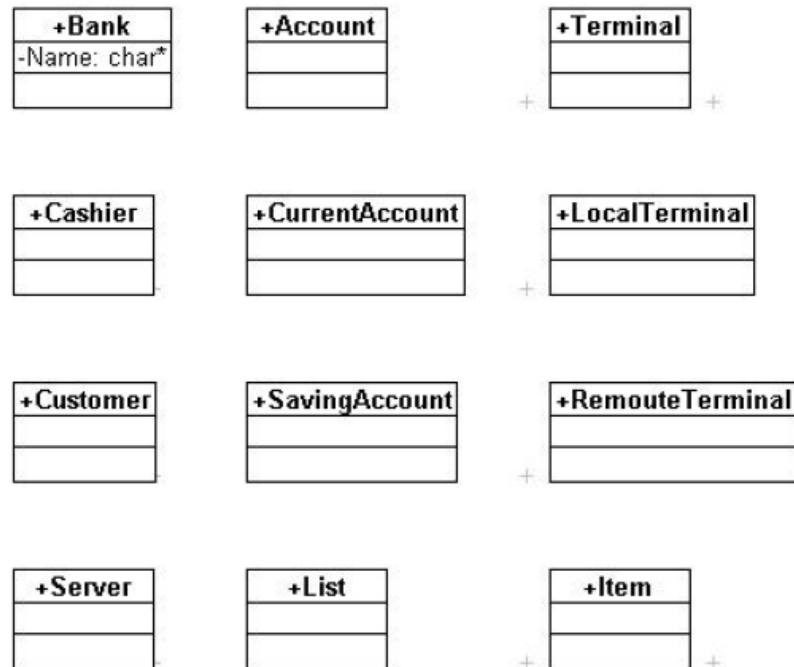


Рисунок 2.1 - Список класів

## 2.5. Підготовка словника даних

Наведемо частину словника даних, яка містить визначення класів, використовуваних в проекті.

Банкомат - термінал, який дає можливість клієнту здійснювати свою власну проводку, використовуючи для ідентифікації свою картку. Банкомат посилає інформацію про проведення в банк для її перевірки. Надалі видає квитанцію клієнту і якщо проводка коректна - видає гроші.

Банк - фінансова організація, яка містить рахунки своїх клієнтів і випускає картки, що санкціонують доступ до рахунків через мережу терміналів.

Картка - пластикова картка, вручена банком своєму клієнтові, яка санкціонує доступ до рахунків через мережу терміналів. Кожна картка містить код банку, номер рахунку та номер картки, закодовані відповідно до

національними стандартами на банківські картки. Картка не обов'язково забезпечує доступ до всіх рахунків клієнта. Кожної картою може володіти тільки один клієнт, але у неї може існувати кілька копій.

Касир - службовець банку, який має право здійснювати проводки, інакше кажучи, може приймати і видавати гроші і чеки клієнтам.

Касовий термінал - термінал, з якого касир здійснює проводки для клієнтів. Коли касир приймає і видає гроші та чеки, касовий термінал друкує квитанції. Касовий термінал взаємодіє з комп'ютером банку, щоб перевірити і виконати проводку.

Клієнт - власник одного або декількох рахунків у банку. Клієнт може складатися з одного чи кількох осіб, або організацій.

Комп'ютер банку - комп'ютер, що належить банку, який взаємодіє з мережею терміналів.

Проводка - одиничний інтегрований запит на виконання деякої послідовності операцій над рахунками одного клієнта. Всі проводки повинні протоколюватися і правильно враховуватися.

Рахунок - одиничний банківський рахунок, над якими виконуються проводки. Рахунки можуть бути різних типів; клієнт може мати кілька рахунків.

## **2.6. Визначення залежностей (зв'язків) між класами**

Необхідно виділити явні і неявні дієслівні обороти з попередньої постановки задачі (опис предметної області) і розглянути їх як імена можливих залежностей.

Дієслівні обороти (явні і неявні):

Банківська мережа включає касирів і термінали

Банк володіє комп'ютером банку

У банку служать касири

Комп'ютер банку підтримує рахунки

Банк володіє всіма терміналами

Касовий термінал взаємодіє з комп'ютером банку

Касир вводить проводку над рахунком

Термінали взаємодіють з комп'ютером банку під час проводки

Термінал приймає картку

Термінал спілкується з користувачем

Термінал видає готівку

Термінал друкує квитанції

Банк надає програмне забезпечення

Система забезпечує протоколювання

Система забезпечує безпеку

Система регулює колективний доступ

Клієнти мають картки

Картка забезпечує доступ до рахунку

Потім необхідно виключити непотрібні або неправильні залежності:

- видаляємо залежності між виключеними класами.

- видаляємо нерелевантні залежності і залежності, пов'язані з реалізацією.

- розбиваємо складні залежності на прості залежності.

- видаляємо похідні залежності.

Таким чином, видаливши надлишкові залежності, отримаємо наступний список залежностей:

Банк володіє комп'ютером банку

Комп'ютер банку підтримує рахунки

Банк володіє касовими терміналами

Термінал взаємодіє з комп'ютером банку

Касир використовує касовий термінал

Касир вводить проводку

Проводка відноситься до рахунку

Банкомат ініціює проводку

Клієнти мають картки

Клієнти використовують банкомати

Картка забезпечує доступ до рахунку

У банку служать касири

Внести зміни і додати невраховані залежності можна буде пізніше, при розробці діаграми класів. Приклад діаграми класів представлений на малюнку 2.2.

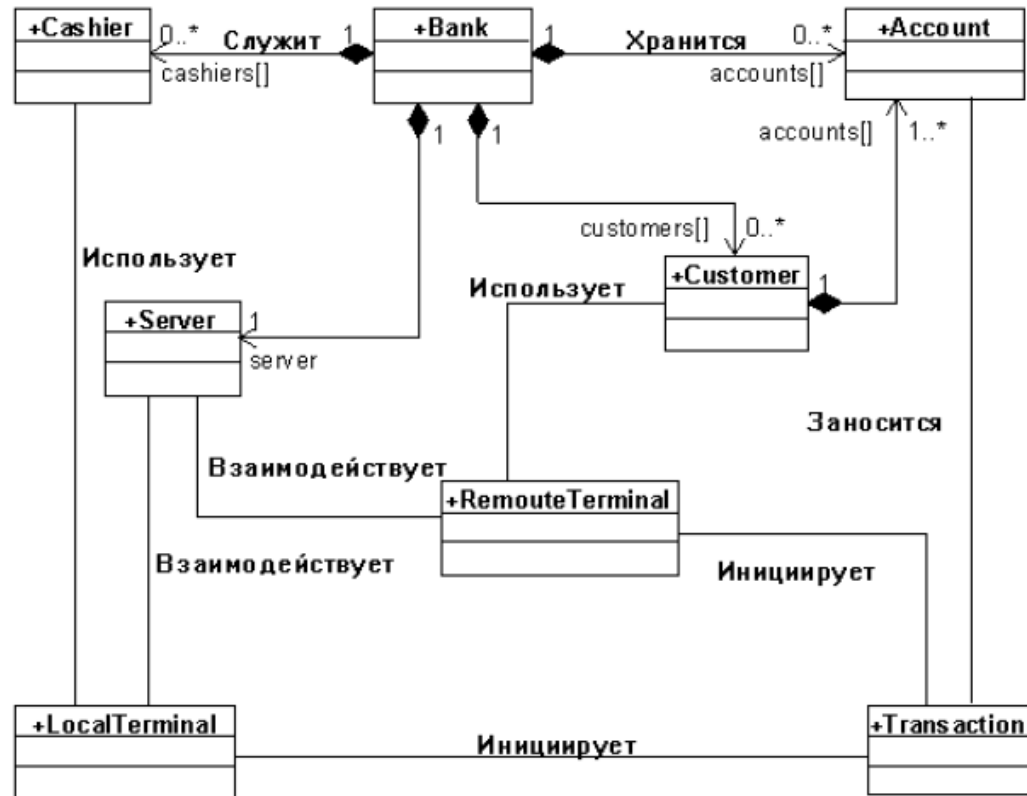


Рис. 2.2. - Діаграма класів

## 2.7. Побудова діаграми ієрархії класів.

Повівши аналіз залежностей між класами на предмет виявлення зв'язків типу «це є узагальненим поняттям того» можна побудувати діаграму успадкування (ієрархію) класів, представлену на малюнку 2.3.

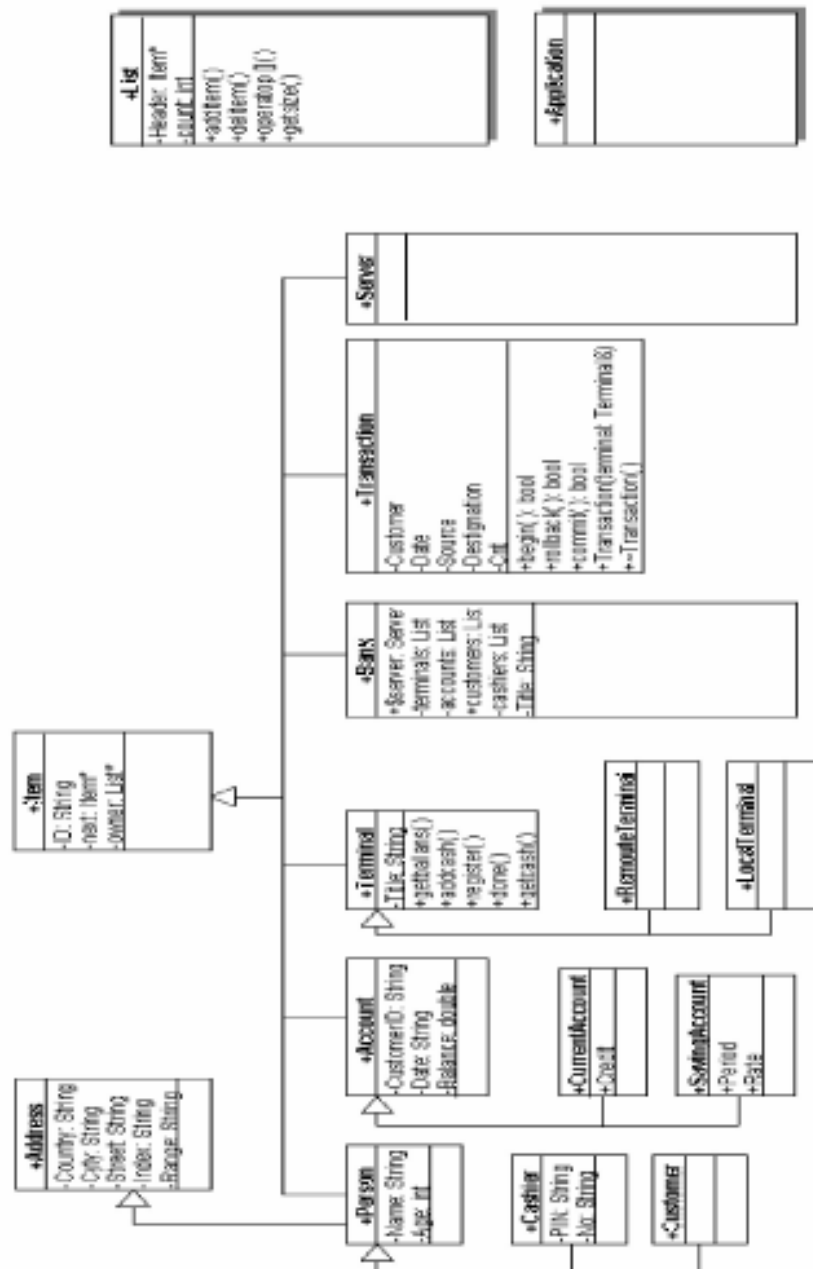


Рис. 2.3. - Діаграма ієрархії класів

### 3. ВАРІАНТИ ЗАВДАНЬ НА КУРСОВУ РОБОТУ

Незалежно від теми програма, що розробляється, повинна задовольняти наступним загальним вимогам.

1. *Стійкість програми.* Програма не повинна втрачати працездатності ні при яких, навіть некоректних, діях користувача. Всякі дії, що загрожують втратою інформації, мають бути підтверджені користувачем. Інформація, що вводиться, скрізь, де це можливо, піддається логічному контролю.

2. *Забезпечення цілісності даних.* При будь-яких діях користувача не повинні втрачатися дані або їх цілісність (некоректність індексів, втрата посилань в зв'язках після видалення-додавання записів і т. д.).
3. *Функціональна повнота.* Мають бути реалізовані усі функції, вказані в специфікації програми.
4. *Терміни і інтерфейс.* У діалогових вікнах використовуються тільки терміни, зрозумілі користувачеві, і не використовуються терміни розробника («запис», «індексація» і т. д.). Поява службових англomовних повідомлень неприпустимо. У повідомленнях користувача слід дотримуватися норм ввічливості, колірна гамма повинна наслідувати загальноприйняті рекомендації.
5. *Використання клавіатури.* На будь-якому етапі натиснення будь-якої клавіші повинне ігноруватися або викликати передбачені дії, описані в засобах допомоги. Прив'язка дій до клавіш має бути загальноприйнятою: F1 - допомога; Enter - згода, завершення введення; Esc - відмова, повернення до попереднього вузла гілки алгоритму (з відновленням екранної форми); Tab - перехід до наступного поля, вікна і т. д.; Shift - Tab - повернення до попереднього поля і т., д.

Усі приклади тем узяті з книги [1].

1. Особиста бібліотека. Картотека домашньої бібліотеки: вихідні дані книги (Автори, назва, видавництво і т.д.), розділ бібліотеки (спеціальна література, хобі, домашнє господарство, белетристика і т.д.), походження і наявність книги в даний момент, суб'єктивна оцінка книги. Вибір книг по безпідставному замовленню; інвентаризація бібліотеки.

2. Картотека Інтерполу. Дані по кожному зареєстрованому злочинцю: прізвище, ім'я, кличка, зріст, колір волосся і очей, особливі прикмети, громадянство, місце і дата народження, останнє місце проживання, знання Зиков, злочинна професія, остання справа і т.д. Злочинні і мафіозні угруповання (дані про співучасників). Вибірка по будь-якій підмножині ознак. Перенесення «Зав'язали в архів; видалення лише після смерті.



3. Бюро знайомств. База потенційних наречених: стаття, реєстраційний номер, дата реєстрації, відомості про себе, вимоги до партнера. Вибір підмножини підходящих кандидатур, підготовка зустрічей (формування запрошення для знайомства). Перенесення в архів пар, які вирішили свої сімейні проблеми, видалення клієнтів, що відмовилися від послуг.

4. Біржа праці. База безробітних: анкетні дані, професія, освіта, місце і посада останньої роботи, причина звільнення, сімейний стан, житлові умови, контактні координати, вимоги до майбутньої роботи. База вакансій: фірма, посада, умови праці та оплати, житлові умови, вимоги до фахівця. Пошук і реєстрація варіантів з того та іншого боку, формування оголошень для друку, видалення в архів після працевлаштування, повне видалення при відмові від послуг.

5. Записна книжка. Анкетні дані, адреси, телефони, місце роботи або навчання, посаду знайомих, колег і родичів, характер знайомства, ділові якості і т.д. Автоматичне формування поздоровлення з днем народження (за поточної дати). Впорядкування за алфавітом і за датою останнього коригування. Пошук по довільному шаблону.

6. Каса Аерофлоту. Розклад: номер рейсу, маршрут, пункти проміжної посадки, час відправлення, дні польоту. Кількість вільних місць на кожному рейсі. Вибір найближчого рейсу до заданого пункту (при наявності вільних місць), оформлення заданого числа квитків за погодженням з пасажиром (з зменшенням числа вільних місць), оформлення посадкової відомості.

7. Довідник споживача (служба побутової). База підприємств побутового обслуговування міста: назва, розряд, адреса та телефони, спеціалізація, перелік надаваних послуг, форма власності, години і дні роботи. Пошук підприємств по заданій послугі та іншими ознаками.

8. Довідник покупця. Бази торгових точок міста: назва, адреса і телефони, спеціалізація, форма власності, час роботи. Вибір магазинів за довільним шаблоном.

9. Магазин з одним продавцем. Комп'ютер замість касового апарату. База наявності товарів: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. Реєстрація надходження товару (як старих, так і нових найменувань). Оформлення покупки: виписка чека, коректування бази. Проблема уцінки і списання. Інвентаризація залишків товару з обчисленням сумарної вартості.

10. Відділ кадрів. База даних про співробітників фірми: паспортні дані, освіта, спеціальність, підрозділ, посада, оклад, дати надходження у фірму і останнього призначення і т.д. Вибір за довільним шаблоном. Скорочення штатів: вибір для звільнення осіб пенсійного та передпенсійного віку, підготовка наказу.

11. Генеалогічне дерево. Паспортні дані членів деякого родового клану; посилення на дітей (або на батьків). Пошук всіх нащадків або всіх предків для вказаної особи.

12. Склад. База товарів, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Висновок інвентарної відомості.

13. Каса автовокзалу. Розклад автобусів: номер рейсу, кінцевий і проміжний пункти, час відправлення. Кількість вільних місць на кожному рейсі. Вибір найближчого рейсу до заданого пункту (при наявності вільних місць), оформлення квитків, оформлення посадкової відомості. Попередній продаж, повернення квитків.

14. Адміністратор готелю. Список номерів: клас, число місць. Список гостей: паспортні дані, дати приїзду і від'їзду, номер. Поселення гостей: вибір відповідного номера (за наявності вільних місць), реєстрація, оформлення квитанції. Від'їзд: вибір всіх постояльців, що від'їжджають сьогодні, звільнення місця або оформлення затримки з випискою додаткової квитанції. Можливість дострокового від'їзду з перерахунком. Пошук гостя по довільною ознакою.

15. Довідник меломана. База груп і виконавців: база пісень: база дисків з переліком пісень (у вигляді посилань). Вибір всіх пісень заданої групи; всіх дисків, де зустрічається задана пісня.

16. Щоденник. База намічуваних заходів - дата, час і протяжність, місце проведення. Автоматичне нагадування найближчої справи: за поточною датою і часу: видалення вчорашніх справ або перенесення на майбутнє. Аналіз "накладок" - перетинань планованих справ. Перегляд справ на завтра, післязавтра і т.д.

17. Термінологія. База визначень будь-якої науки: вводиться термін, його тлумачення (визначення), посилання на використовувані терміни. Можливість перегляду всього ланцюжка від заданого терміна до первинних понять.

18. Шеф-кухар. База рецептур страв: розкладка, рецепт приготування. База продуктів на складі: найменування, ціна, кількість. Формування меню на день (на задане число осіб); звану вечерю. Перевірка достатності запасів: формування видаткової накладної на склад, коригування запасів.

19. Довідник лікаря. База хвороб: назва, симптоми, процедури, перелік рекомендованих ліків із зазначенням необхідної кількості. База медикаментів на складі: назва, кількість, взаємозамінність. Формування рецепта після огляду хворого, перевірка наявності ліків, коригування запасів.

21. Довідник фірм. Назва, адреса та телефони, перша особа, статус (форма власності), сировина, продукція. Вибір за довільним шаблоном.

22. Обмін житла. База пропозицій по обміну: район, площа, планування і т.д.; вимоги до варіантів обміну. Реєстрація клієнтів, вибір відповідних варіантів, видалення при відбулося обміні або відмову.

23. Справочник абитуриента. База вузов; наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм), размер оплаты при договорном обучении. Выбор по разным критериям: все о данном вузе; все о данной специальности, поиск минимального конкурса по данной специальности или

вообще. 20. Зарахування абітурієнтів. База абітурієнтів; анкетні дані, сукупність оцінок на вступних іспитах, готовність вчитися на договірній основі. Вибір для зарахування заданої кількості абітурієнтів; формування для співбесіди списку тих, хто набрав граничний прохідний бал, але не може платити за освіту.

24. Довідник поштової індексації. Республіка, область (край), район, населений пункт, поштовий індекс. Пошук по будь-якої сукупності полів (крім останнього); ієрархічна зв'язок між полями (зверніть увагу, що, наприклад, Новомосковськ є і в Тульській, і в Дніпропетровській областях).

25. Ощадбанк. Відомості про вкладників банку: номер особового рахунку, категорія вкладу, паспортні дані, поточна сума внеску, дата останньої операції. Операції прийому і видачі будь-якої суми, автоматичне нарахування відсотків. 26. Ломбард. База збережених товарів та нерухомості: анкетні дані клієнта, найменування товару, оціночна вартість; сума, видана під заставу, дата здачі, термін зберігання. Операції прийому товару, повернення, продажу - по закінченню терміну зберігання.

27. Довідник селекціонера. Найменування сорту будь-якої культури, автор, батьківські сорти, врожайність, характеристики плодів, морозостійкість, стійкість до шкідників і хвороб, наявність в тому чи іншому фонді. Вибір сортів, що володіють заданими властивостями.

28. Довідник власника відеотеки. База відеофільмів: назва, студія, жанр, рік випуску, режисер, виконавці головних ролей, короткий зміст, суб'єктна оцінка фільму. Факт наявності фільму у відеотеці. Оформлення видачі та повернення касети.

29. Купи - продай. База продавців: найменування товару, обсяг партії при оптовому продажі, ціна, умови продажу - відвантаження, форма оплати, контактну адресу або телефон, примітка (наприклад, «посередників прошу, не турбується»). База покупців: найменування товару, обсяг покупки, прийнятна ціна і форма оплати, контактну адресу або телефон, примітка. Пошук і реєстрація варіантів з тієї та іншої сторони; формування оголошень

для друку, видалення в архів після купівлі - продажу (можливо, один із клієнтів залишається незадоволеним), повне вилучення при відмові від послуг.

30. Довідник фаната. База спортсменів: анкетні і антропологічні дані, громадянство, походження, вид спорту, клуб або команда, дані про особовий рекорді або перемоги і так далі. Вибір за довільним ознакою. Пошук рекордсмена в заданому виді спорту.

31. Довідник радіоаматора. Бази довідкових даних транзисторів, діодів, тиристорів і так далі: марка, характеристики, гранично допустимі умови експлуатації, ціна, облік взаємозамінності і т.д. Підбірка по заданим вимогам.

32. Довідник комерційних банків. Найменування, адресу, статус (форма власності), умови зберігання коштів на особовому рахунку (річні відсотки на різних видів вкладів). Вибір банку з найбільшим відсотком для заданого типу вкладів.

33. Довідник начальника в'язниці. Анкетні дані ув'язнених, стаття, термін, дата взяття під варту, місце в тюремній ієрархії, камера, відомості про родичів, особливості характеру. Формування статистичних зведень про склад, вибір за довільною ознакою.

34. Довідник командира. Список підлеглих військовослужбовців: анкетні дані, адресу батьків, цивільна професія, освіта, звання та дата його отримання, посада, підрозділ, форма служби, особливості характеру і ставлення до служби. Формування списків: заданого підрозділи, офіцерського складу, новобранців і т.д. 35. Ріелтерська контора (купівля-продаж житла). База пропозицій: район та адреса, характеристика будинку і квартири, запитувана вартість, координати заявника. База попиту: вимоги покупця до житла (можливо кілька варіантів, допустимі діапазони), фінансові можливості, координати заявника. Підбір варіантів для того чи іншого боку, автоматизований пошук взаємоприйнятних варіантів. Приклад запиту покупця: однокімнатна, до 100 тис. руб., Інкерман не пропонувати.

36. Черга на житло. Список черговиків на отримання і поліпшення житла: дата надходження у фірму, дата подачі заяви, склад сім'ї, житлові умови на дату подання заяви, пільги на додаткову площу, позачергове і першочергове поліпшення, побажання на район і ін Реалізація розподілу одержуваного муніципального житла: задоволення підходящих черговиків, переселення наступних в звільнилися квартири. Додавання заявників та видалення вибулих і задоволених.

37. Автосалон. База нових і старих вітчизняних та іноземних автомобілів: марка, рік випуску, технічні характеристики, особливості виконання, технічний стан, запитувана ціна. База покупців: контактні координати, вимоги до марки, технічним характеристикам і технічним станом, фінансові можливості. Автоматизація підбору варіантів для покупця, формування заявки для постачальників і перегонників.

38. Довідник туриста. Туристичні агенції та пропоновані послуги: країна, місто (або маршрут круїзу), умови проживання та проїзду, екскурсійне обслуговування, сервіс прийнятої боку, вартість путівки.

39. «Купи-продай». База оголошень (радіо-, газетних, в біжучому рядку) по всіх рубриках. Пошук для ... (наприклад, «репетиторство & математика»).

40. Крилаті фрази. Довідник прислів'їв, приказок, афоризмів, каламбурів, інших словесних курйозів. Класифікація за авторам і джерелам, пошук по темах і ключовим словам.

41. Каталог запчастин автомобіля. В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках. Таблиці: країна, фірма-виробник, марка автомобіля, агрегат, вузол, деталь. Облік взаємозамінності. Користувачі: працівники автосервісу, магазинів запчастин; постачальники-оптовики. 42. Каталог радіодеталей (довідник телевізійного обладнання). Моделі побутової аудіо-, та відеотехніки; для кожної моделі - каталог радіодеталей, використаних в ній (резистори, конденсатори, котушки індуктивності, чіпи тощо). Багато деталей використовуються в різних

моделях; деякі деталі однобічно (наприклад, замість резистора 10К 0,25 Вт, але не навпаки) або двостороннє (транзистори з близькими характеристиками) замінювані. Облік наявності деталей на складі, пошук відповідних деталей для ремонту.

43. Довідник нумізмата. Монети: країна, номінал, рік випуску, кількість випущених монет, особливості. Колекціонери: країна, ім'я, контактні координати, наявність рідкісних монет в колекції. Власна колекція.

44. Довідник філателіста. Марки: країна, національна вартість, рік випуску, тираж, особливості. Філателісти: країна, ім'я, контактні координати, наявність рідкісних марок в колекції. Власна колекція.

45. Довідник любителя живопису. Художники з анкетними даними та стилями. Картини з посиланням на художників, датою створення, жанром. Колекціонери та музеї: наявність оригіналів картин і копій. Аукціонери і комісіонки: дата проведення, список виставлених шедеврів і ціни на них. Власна колекція.

46. Так-тіклі. Ця гра схожа на шашки і на хрестики нулики. Ігрове поле - це 16 клітин (4x4). Кожен з гравців ставить свої чотири фішки по обидва боки поля, через одну з фішками противника, як показано на малюнку. За один хід можна пересунути свою фішку на одну вільну клітину вгору або вниз, вправо або вліво, але не по діагоналі. Знімати фішки супротивника не можна. Завдання полягає в тому, щоб розташувати три фішки свого кольору в один ряд по вертикалі, горизонталі або діагоналі швидше противника.

47. Хід конем. Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають коня. Напрямок ходу довільно, але має відповідати правилам руху шахового коня. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

48. Хід слоном. Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають слона. Напрямок ходу довільно, але має

відповідати правилам руху шахового слона. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

49. Хід королеви. Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають ферзя. Напрямок ходу довільно, але має відповідати правилам руху шахового ферзя. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

50. Хід тури. Мета гри: набрати більше очок, ніж супротивник. Правила гри: Гравці по черзі рухають туру. Напрямок ходу довільно, але має відповідати правилам руху шахової тури. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

51. Хід амазонки. Мета гри: набрати більше очок, ніж суперник. Правила гри: Амазонка - уявна <шахова> фігура, яка об'єднує властивості ферзя і коня. Гравці по черзі рухають амазонку. Напрямок ходу довільно, але має відповідати зазначеним правилам. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

52. Шибениця. Мета гри: вгадати слово, загадане комп'ютером. Правила гри: Можна вибрати будь-яку літеру. Якщо така буква є, всі аналогічні букви в слові будуть відкриті.

53. Решітка. Ігрове поле має розмір 10x10 з випадково розподіленими числами в них. Мета гри: Набрати більше очок, ніж супротивник. Правила гри: Перший гравець ходить по горизонталі, другий - по вертикалі. Перший гравець може вибрати будь-яку клітину з цифрою на поточному горизонталі.



Якщо колір клітини червоний, то кількість очок, вказане на ній, додається до рахунку гравця, якщо синій - віднімається. Після ходу першого гравця вертикаль, на якій знаходиться вибрана ним клітина, стає поточною і хід переходить до супротивника, який може вибрати будь-яку клітину з цифрою на поточному вертикалі і т.д.

54. Діагоналі. Ігрове поле має розмір 6х6. Мета гри: Зайняти якомога більше місця на дошці. Правила гри: Коли настає Ваша черга ходу, Ви можете поставити одну фішку на будь-яку клітину дошки. Якщо після Вашого ходу, один чи кілька фішок супротивники опинилися між вашими фішками, все етіфішки суперника переходять до Вас.

55. Double-dealing fellows. Ігрове поле має розмір 6х6. Мета гри: зайняти на дошці простір більше, ніж супротивник. Правила гри: Можна рухати яку свою фішку на наступну клітину дошки, при цьому фішка розділиться на двоє, тобто з'явиться в новій позиції і залишиться на старій одночасно. Також можна перестрибувати через одну клітку, але в цьому випадку <роздвоєння> фішки не відбувається. Якщо сусідні клітини зайняті суперником, вони переходять до Вас.

56. Сенегальські шашки. Розмір ігрового поля - 5х6 клітин. Кожен гравець отримує по 14 шашок (черепашки і каміння), які розставляються так, щоб заповнилися дві крайні верхні і нижні лінії шашками відповідного кольору. У центрі шашки розміщаються з країв, так щоб утворилося два порожніх поля, з яких починається гра. Рухатися можна тільки на порожнє поле попереду, праворуч або ліворуч, по одній клітці за хід. Не можна рухатися назад або по діагоналі. Якщо з'являється можливість перестрибнути через шашку супротивника, гравець зобов'язаний зробити це. У такому випадку, шашка супротивника вважається <убитою> і знімається з дошки. Гра закінчується, коли шашки одного з партнерів просунулися так далеко, що більше не можуть <з'їсти> фішок. Тоді підраховується кількість з'їдених шашок і визначається переможець.

57. Японські шашки. Ігрове поле має розмір 10x10. Мета гри: поставити 5 фішок в ряд (по вертикалі, горизонталі, діагоналі). Правила гри: Фішку можна ставити на будь-який вільний місце ігрового поля.

58. Куточки. Ігрове поле має розмір 7x7. Спочатку фішки розташовуються двома блоками 3x3 в протилежних кутках дошки. Мета гри - перемістити всі фішки в протилежний кут. Шашку можна пересувати двома способами: на одну клітку вліво, вправо, вгору або вниз, або перестрибувати через інші фішки.

59. Блокада. Ігрове поле має розмір 7x7. Спочатку фішки розташовуються двома блоками 3x3 в протилежних кутках дошки. Мета гри: пересунути всі шашки в протилежний кут швидше противника. Правила гри: Через порожні сусідні поля можна просувати фішку до трьох клітин за хід. Можна перестрибувати, через фішку. У цьому випадку фішка, через яку гравець перестрибнув, відправляється <в дім> - в кут, з якого вона почала свій рух. Кожен гравець може зробити не більше 10 стрибків за гру.

#### **4.ВИМОГИ ДО ЗМІСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ**

Пояснювальна записка повинна містити наступні розділи:

- вступ. Даний розділ повинен містити інформацію про призначення та галузі застосування програми.

- постановка задачі. Даний розділ повинен включати вихідні дані по варіанту завдання, розгорнутий опис предметної області, термінологічний словник.

- опис програми. Даний розділ повинен включати опис системи, що моделює предметну область, в термінах об'єктно-орієнтованого підходу (діаграми ієрархії класів і об'єктів та їх опис), приклад тестових вхідних і вихідних даних.

- керівництво користувача

- висновки

- бібліографія. Даний розділ повинен включати джерела, використані при розробці.

- додатки. Програми повинні містити програмний код розроблених класів.)

Залежно від особливостей документа окремі розділи (підрозділи) допускається об'єднувати, а також вводити нові розділи (підрозділи).

## **ВИМОГИ ДО ФОРМИ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ**

### ***Загальні вимоги***

Загальними вимогами до пояснювальної записки є логічна послідовність викладання матеріалу, стислість, чіткість і конкретність викладання теоретичних і практичних результатів роботи, доказовість висновків і обґрунтованість рекомендацій.

Записку виконують із розрахунку не більше 40 рядків на сторінці за умови рівномірного її заповнення і висотою літер і цифр не менше 1,8 мм. При написанні тексту необхідно дотримуватись таких розмірів полів: верхнє, ліве і нижнє - не менше 20 мм, праве - не менше 10 мм.

Прізвища, назви додатків, організацій, фірм та інші імена наводяться мовою оригіналу.

Пояснювальна записка виконується машинним способом на одній стороні аркушу білого паперу формату А4 (210 x 297 мм) відповідно до ДСТУ 3008-95 "Документація. Звіти в сфері науки і техніки. Структура і правила оформлення" [2]. Наведемо деякі положення цього документу.

### ***Структурні елементи та розділи***

Треба розрізняти структурні елементи записки, такі як "РЕФЕРАТ", "ЗМІСТ", "ВСТУП", "СУТЬ" і розділи записки. Деякі елементи, а саме "РЕФЕРАТ", "ЗМІСТ", "ВСТУП", "ВИСНОВКИ", "ПЕРЕЛІК ПОСИЛАНЬ", створюють розділи записки і їхні найменування служать заголовками окремих розділів. Такі розділи не мають нумерації. Структурний елемент записки „СУТЬ” сам може поділятися на розділи, які нумеруються и мають власні заголовки.

Розділи можуть поділятися на підрозділи, пункти та підпункти.

### ***Заголовки***

Розділи і підрозділи *повинні* мати заголовки, пункти і підпункти *можуть* мати заголовки. Заголовки структурних елементів записи і заголовки розділів необхідно розташовувати посередині рядка і друкувати прописними літерами без крапки наприкінці, не підкреслюючи. Заголовки підрозділів, пунктів та підпунктів записи необхідно починати з абзацного відступу і друкувати малими літерами, крім першої прописної, не підкреслюючи, без крапки наприкінці. Абзацний відступ повинен бути однаковим по всьому тексту записи і має дорівнювати п'яти знакам. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою.

Переноси слів у заголовку розділу не припускаються.

Відстань між заголовком і наступним або попереднім текстом повинна бути не менше двох рядків.

Не припускається розміщати найменування розділу, підрозділу, а також пункту і підпункту в нижній частині сторінки, якщо після нього розташований тільки один рядок тексту.

### ***Нумерація сторінок***

Нумерувати сторінки необхідно арабськими цифрами, дотримуючи наскрізну нумерацію по всьому тексті. Номер сторінки проставляють у правому верхньому кутку без крапки наприкінці.

### ***Нумерація розділів та підрозділів***

Розділи, підрозділи, пункти, підпункти пояснювальної записки варто нумерувати арабськими цифрами. Розділи пояснювальної записки повинні мати порядкову нумерацію в межах суті пояснювальної записки і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3 . Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Нумери підрозділів складаються з номерів розділів і підрозділів, що розділяються крапкою, наприклад, 1.1, 1.2, 1.3 .

## ***Переліки***

За необхідністю можуть бути використані переліки. Перед переліком ставлять двокрапку. Перед кожною позицією переліку треба ставити малу літеру алфавіту з дужкою або дефіс (перший рівень деталізації). Для подальшої деталізації переліку треба використовувати арабські цифри з дужкою (другий рівень деталізації).

Приклад:

- а) перша версія бібліотеки;
- б) друга версія бібліотеки:
  - 1) для персональних комп'ютерів;
  - 2) для мобільних пристроїв;
- в) третя версія бібліотеки.

## ***Ілюстрації***

Ілюстрації треба розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці. На всі ілюстрації повинні бути посилання в пояснювальній записці. Ілюстрації можуть мати назву, що розташовують під ілюстрацією. При необхідності під ілюстрацією ще розташовують пояснювальні дані. Ілюстрація позначається словом “Рисунок”, що разом із назвою поміщають після пояснювальних даних, наприклад, “Рисунок 2.1 - Схема алгоритму”. Ілюстрації необхідно нумерувати арабськими цифрами в межах розділу, за винятком ілюстрацій, наведених у додатках. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Якщо ілюстрація не вміщується на одній сторінці, можна переносити її на інші сторінки, при цьому назву ілюстрації розташовують на першій сторінці, пояснювальні дані - на кожній сторінці, і під ними вказують: “Рисунок , \_\_\_ аркуш ”.

## ***Таблиці***

Таблиці необхідно розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на

наступній сторінці. На всі таблиці повинні бути посилання в пояснювальній записці. Таблиця може мати назву, яку друкують малими літерами (крім першої прописної) і розміщують над таблицею. Якщо рядки або графи таблиці виходять за формат сторінки, таблицю поділяють на частини, розташовуючи одну частину під іншою, або поруч, або переносячи частини таблиці на наступну сторінку. Слово “Таблиця \_\_\_\_”, вказують один раз зліва над першою частиною таблиці, над іншими частинами пишуть: “Продовження таблиці \_\_\_\_\_” з вказівкою номера таблиці. Таблиці треба нумерувати арабськими цифрами в межах розділу, за винятком таблиць, наведених у додатках. Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою.

### ***Формули і рівняння***

Формули і рівняння розташовують у пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, посередині сторінки. Вище і нижче кожної формули повинно бути залишено не менше одного вільного рядка. Формули і рівняння треба нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком формул і рівнянь, наведених у додатках. Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, розділених крапкою. Номер формули або рівняння вказують на рівні формули або рівняння в дужках у крайньому правому положенні на рядку. Пояснення символів і числові коефіцієнти, що входять до формули або рівняння, варто робити безпосередньо після формули або рівняння в тій послідовності, у якій вони надані у формулі або рівнянні. Пояснення значення кожного символу або числового коефіцієнта необхідно давати з нового рядка. Перший рядок пояснення починають з абзацу словом “де” без двокрапки. Переносити формули або рівняння на інший рядок припускається тільки на знаках виконуваних операцій, причому знак операції на початку такого рядка повторюють. При переносі формули або рівняння на знаку операції

множення застосовують знак “х”. Формули, що прямують одна за одною, і не розділені текстом, відокремлюють комою.

### ***Посилання***

Посилання в тексті на джерела необхідно наводити в послідовності їх згадування в записці, вказуючи порядковий номер, виділений двома квадратними дужками, наприклад: [5].

### ***Скорочення***

У тексті пояснювальної записки припускається робити тільки загальноновживані скорочення (наприклад "і т.д., і т.п., та ін.") або поширені аббревіатури. Якщо в пояснювальній записці прийнята специфічна термінологія, або вживаються малопоширені скорочення, нові символи, позначення і ін., їх перелік необхідно надати в окремому списку. Перелік повинен розташовуватися стовпчиком, у якому слова за абеткою наводяться скорочено, а справа - їхнє докладне тлумачення. Якщо спеціальні символи, терміни, скорочення, і позначення наводяться менше трьох разів, перелік їх не складається, а їхнє тлумачення роблять у тексті при першому згадуванні.

### ***Перелік посилань***

Перелік посилань оформлюється як розділ пояснювальної записки з нової сторінки відповідно до вимог ГОСТ 7. 1-84 "Библиографическое описание документа. Общие требования и правила составления".

Для оформлення бібліографічних посилань, правильного запису загальноприйнятих скорочень рекомендується використовувати методичні вказівки [3], які складені з урахуванням відповідних стандартів.

### ***Додатки***

Додатки необхідно оформляти як продовження пояснювальної записки на наступних її сторінках, або в окремій частині, розташовуючи додатки в порядку появи посилань на них у тексті пояснювальної записки. Кожний додаток повинен починатися з нової сторінки і мати заголовок, надрукований вгорі малими літерами з першої прописної, симетрично щодо тексту

сторінки. Посередині рядка над заголовком малими літерами з першої прописної повинно бути надруковано слово “Додаток \_\_\_\_” і прописна буква, що позначає додаток. Додатки слід позначати послідовно прописними буквами українського алфавіту, за винятком букв Ї, Є, З, І, Ї, Й, О, Ч, Ь. Один додаток позначається як А. Додатки повинні мати загальну з іншою частиною пояснювальної записки наскрізну нумерацію. Текст кожного додатка, при необхідності, може бути розділений на підрозділи і пункти, що нумеруються арабськими цифрами в межах кожного додатка.

## **5. ЛІТЕРАТУРА**

1. Бадд Т. Объектно-ориентированное программирование в действии: Пер. с англ./Тимоти Бадд. – СПб.: Питер, 1997.– 464 с.
2. Бондарев В.М., Программирование на С++. Учебное пособие. – Харьков: СМІТ, 2004г. – 294 с.
3. Гамма Э. И др. Приемы объектно-ориентированного программирования. - СПб. :Питер, 2001г.
4. Дейтел Х., Дейтел П. Как программировать на С++. Третье издание.- М.:ЗАО „Издательство БИНОМ”, 2001г.-1152с.
5. Дьюхарт С. Старк К. Программирование на С++. Киев: Диалог Софт, 1993г.
6. Ирэ Пол. Объектно-ориентированное программирование с использованием С++: Пер. с англ. - Киев: НИИПФ ДияСофт Лтд, 1995. 480с.
7. Павловская Т.А. С/С++. Программирование на языке высокого уровня.- СПб.: Питер, 2001.- 464 с.
8. Прата Стивен. Язык программирования С. Лекции и упражнения. Учебник.- СПб.: ООО «ДиаСофтЮП», 2002.- 896 с.
9. Складов В.А. Язык С++ и объектно-ориентированное программирование: Справочное издание. - Минск: Вышэйшая школа, 1997. 480с.



10. Страуструп Б. Язык программирования С++. Специальное издание.-М., СПб.: "Издательство БИНОМ" – „Невский Диалект”, 2001г.-1099 с.
11. Сэвитч У. С++ в примерах.-М:Экон, 1997г.
12. Фридман А.Л. Основы объектно-ориентированной разработки программных систем. -М. Финансы и статистика. 2000г.
13. Черепанцев А.С. Введение в С и С++. Таганрог.
14. Шмак Н.К. Основы С++ и объектно-ориентированного программирования. Киев: Диалектика, 1996г.

### **ПЕРЕЛІК ПОСИЛАНЬ**

1. Юркин А.Г. Задачник по программированию. – СПб.: Питер, 2002. – 192 с.
2. ДСТУ 3008-95 "Документація. Звіти в сфері науки і техніки. Структура і правила оформлення" – Державний стандарт України.
3. Методические указания по оформлению библиографических ссылок и списков к курсовым, дипломным и научным работам для студентов всех форм обучения и всех специальностей /Сост. Л.П. Гуданова, А.П.Мальцев, Л.П.Щербакова, А.К.Безуглая. - Харьков: ХИРЭ, 1986. - 36 с.

## ДОДАТОК А

### Узгодження що до кодування програм

Курсову роботу слід виконувати на мові C# або java script. Нижче викладені вимоги до вихідного коду програми на C #.

#### 1 Структура файлів

##### 1.1 Структура проекту

Кожен проект повинен розташовуватися в окремому підкаталозі каталогу рішення. Ніколи не кладіть файли проекту та файли рішення в один та той же каталог.

Використовуйте підкаталоги для організації вихідного коду проекту. Простіри імен повинні відповідати структурі каталогів.

##### 1.2 Структура файлу вихідного коду

Файл вихідного коду повинен мати наступну структуру (порядок елементів також має значення):

- а) using - декларації;
- б) Декларації простіру імен;
- в) При необхідності переліснелі та допоміжні структури, потрібні для даного класу.
- г) Оголошення класу. Рекомендується оголошувати один клас в одному файлі.

##### 1.3 Структура класів

Класи та структури повинні оголошуватися в наступному порядку:

- а) Поля;
- б) Конструктори;
- в) Властивості (спочатку public, потім protected, потім private);
- г) Методи (спочатку public, потім protected, потім private);
- д) Делегати та події.

#### 2 Форматування коду

Код повинен бути відформатований відповідно до наступних правил:

- а) Параметр "Tabs" повинен бути встановлен в положення "Keep spaces";
- б) Розмір відступу повинен бути 4 пробілу;
- в) Програмні конструкції вибору та повторення повинні бути відформатовані, як в наступних прикладах:

```
if (file.Exists(fileName))
{
    file.Open(fileName);
}
```

```
for (int i = 0; i < MAX_ELEMENTS; ++i)
{
```

```
array[i] = i * MULTIPLY_FACTOR;  
}
```

```
switch (workMode)  
{  
    case WorkMode.Add:  
        DoAdd();  
        break;  
    case WorkMode.Update:  
        DoUpdate();  
        break;  
}
```

г) Порожні рядки використовуються для поліпшення сприйняття коду. Зазвичай вони використовуються для поділу незалежних секцій коду. Обов'язково розміщуйте порожній рядок:

- 1) Після останнього оголошення "using" перед оголошенням простору імен;
- 2) Між методами;
- 3) Між оголошенням локальних змінних методу та першої інструкцією методу;
- 4) Перед багаторядковим та однорядковим коментарем за винятком коментаря після фігурної дужки, що відкриває блок;
- 5) Перед логічно ізольованою частиною коду методу.

д) Пробіли також слід використовувати для поліпшення сприйняття коду. Необхідно вставляти пробіли:

1) Перед круглими дужками, наступними за ключовим словом:  
`if (condition)`

2) Після коми в списку аргументів:  
`int result = Calculate(argumentOne, argumentTwo);`

3) Між бінарним оператором та його операндами:  
`int result = argumentOne + argumentTwo;`

4) Між частинами інструкції "for":  
`for (int i = 0; i < MAX_ELEMENTS; ++i)`

е) Аргументи унарних операторів ніколи не відокремлюються пробілами:  
`i++;  
--counter;`

ж) Всі довгі рядки повинні бути згорнуті. Довжина рядка не повинна перевищувати 80 символів. При згортанні довгих рядків намагайтеся дотримуватися наступних правил:

- 1) Переносите рядок після коми;
- 2) Переносите рядок перед оператором;
- 3) Віддавайте перевагу «логічному» розбиттю рядка, а не «фізичному»;
- 4) Робіть додатковий відступ перед «згорнутої» частиною рядка:

```
int i;  
do  
{  
  
    // Here comes a long line.  
    bool isSucceeded = a + b + c + d + e - h * MAX_FACTOR / DEFAULT_SCALE  
        - correctionFactor;  
}
```

з) Довгий список параметрів методу також повинен бути згорнутий, так само як та список аргументів у інструкції виклику методу:

```
// The method declaration.  
public float DoSomethingFromManyArguments(  
    int argumentOne,  
    int argumentTwo,  
    decimal argumentThree,  
    string argumentFour)  
{  
    // Method body goes here.  
}  
  
// The calling code.  
float result = DoSomethingFromManyArguments(  
    1,  
    1,  
    0.5m,  
    "Some text");
```

и) Не оголошуйте декілька змінних в одному рядку, використовуйте окремий рядок для кожної змінної:

```
int customerID;  
string customerName;  
float salary;
```

### 3 Угоди про найменування

Не використовуйте угорську або іншу префіксних нотацію за винятком випадків, обумовлених далі.

Імена локальних змінних та аргументів методів повинні записуватися в нотації «верблюду»:

```
string customerName;
```

```
private int DoSomething(int firstArgument, float secondArgument);
```

Імена закритих полів класу повинні слідувати нотації «верблюду»:

```
private string customerName;
```

Імена відкритих полів класу повинні слідувати нотації «Паскаль»:

```
public int CustomerID;
```

Імена властивостей та методів класу повинні слідувати нотації «Паскаль»:

```
public bool ValidateAmount();
```

```
public bool FileExists
```

```
{  
    get;  
}
```

Імена класів, структур та нумераторів повинні слідувати нотації «Паскаль»:

```
public class CustomerAccount
```

```
{  
};
```

```
private enum WorkMode
```

```
{  
    Add,  
    Update;  
}
```

Імена методів повинні слідувати шаблоном «дієслово» + «іменник» - наприклад, "UpdateAccount";

Використовуйте однину, а не множину в іменах перелічного. Іншими словами, "WorkMode" - це правильне ім'я, а "WorkModes" - ні;

Використовуйте значущі імена навіть для закритих методів, властивостей, типів та інше. Уникайте використовувати короткі імена, такі як "a", "b", "n" за винятком загальноприйнятих імен для змінних циклу "i" и "j".

#### 4 Загальні принципи розробки

Культура кодування:

а) Дублювання коду в програмі суворо забороняється.

б) Обов'язкова заміна числових та строкових літералів символічними константами. Виняток можна зробити для самоочевидних констант, таких як 0 та 1. Давайте символічним константам значущі імена;

в) Рекомендується об'єднувати пов'язані між собою константи цілих типів в нумератори, а не цілих типів - в абстрактні класи з відкритими статичними константними членами;

г) Уникайте довгих та складних методів, розбивайте їх на декілька коротких. При модифікації коду максимально використовуйте рефакторинг;

д) Завжди звертайте увагу на попередження компілятора. Подбайте, щоб їх не було.

## 5 Угоди про коментарі

Обов'язкові коментарі до класів та до інших призначених для користувача типів та до всіх відкритих членів класів.

Коментарі всередині коду методів не обов'язкові. Код повинен легко читатися та без коментарів.

Коментар не повинен перефразовувати те, що написано в коді. Намагайтеся давати змістовні пояснення. Наприклад, коментар “Increment i by one” до коду “i++;” це погана практика, а коментар “Update the number of customer accounts processed” – добра.

Текст коментаря відокремлюйте від слешів одним пропуском. Перша буква речення повинна бути прописною, в кінці речення повинна стояти крапка:

```
// This is a comment. This is yet another comment sentence.
```

Коментарі повинні підкорятися загальним правилам згортання рядків.

Не використовуйте коментарі в стилі C: /\* ... \*/

## 6 Специфіка Windows Forms

Дотримуйтеся Microsoft User Interface Guidelines, якщо до вас не пред'являють інших вимог.

Якщо два керуючих елемента служать одній меті (наприклад, поле введення для імені користувача та мітка перед ним), дозволяється використовувати тип елемента в закінченні імені. Наприклад, "customerNameText" та "customerNameLabel".

Слід прибирати всю бізнес-логіку з форм та керуючих елементів та розташовувати її в класах моделі. В обробники подій елементів розміщуйте тільки виклики методів.

Використовуйте нотацію «верблюд» для найменування екземплярів керуючих елементів на формах та призначених для користувача керуючих елементах, наприклад, "optionsHeading".

Використовуйте нотацію «Паскаль» для найменування класів форм та керуючих елементів, наприклад, "ProductList".

Прагніть до повторного використання UI та вихідного коду. Делите ваш інтерфейс на елементи; сміливо наслідуйте форми та керуючі елементи.

Джерело інформації:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpgenref/html/cpconnetframeworkdesignguidelines.asp>