

**Перший рівень**

1. Визначити ієрархію класів - побудувати модель програмної системи - відповідно до варіанта (з графічним відображенням у звіті), яка полягає у побудові базового абстрактного класу та кількох обґрунтованих похідних (але не менше трьох) класів. Вибір відношень між класами повинен обґрунтовуватися.
2. Визначити в базовому абстрактному класі чисто віртуальну функцію.
3. Продемонструвати поліморфні виклики методів.
4. Програма використовує три файли: заголовний h-файл із визначенням класу, сpp-файл із реалізацією класу, сpp-файл демонстраційної програми.

**Другий рівень**

1. Визначити ієрархію класів - побудувати модель програмної системи - відповідно до варіанта (з графічним відображенням у звіті), яка полягає у побудові базового абстрактного класу та кількох обґрунтованих похідних (але не менше трьох) класів. Вибір відношень між класами (наслідування одиночне або множинне, відкрите або закрите) повинен обґрунтовуватися.
2. Визначити в базовому абстрактному класі чисто віртуальну функцію.
3. Визначити на базі кожного похідного класу методи, конструктори без параметрів, з параметрами, копіювання (базовий та похідні класи повинні містити не менше п'яти методів). Визначити в частині класів деструктори, а також віртуальні деструктори та продемонструвати відмінність їх роботи.
4. Кожен клас, що входить в ієрархію, що використовує відкрите наслідування, повинен містити принаймні одну віртуальну функцію. Функціональність зазначених функцій не повинна дублюватися.
5. Створити масив об'єктів цих класів.
6. Визначити метод Service, який записує у файл та читає з файлу поля даних класу. Передбачити усі можливі виключні ситуації роботи з файлами (з демонстрацією роботи через try-catch та throw - не менше п'яти - не менше п'яти). **Ім'я створюваного файлу є параметром, що задається в командному рядку (робота програми здійснюється саме з виклику даного методу у функції main() ).**
7. Програма повинна надавати користувачу можливість вибору створення та знищення певного об'єкта (основні сутності) засобами меню.
8. Продемонструвати **поліморфні виклики методів**.
9. Програма використовує три файли: заголовний h-файл із визначенням класу, сpp-файл із реалізацією класу, сpp-файл демонстраційної програми.

**Третій рівень**

1. Визначити ієрархію класів - побудувати модель програмної системи - відповідно до варіанта (з графічним відображенням у звіті), яка полягає у побудові базового абстрактного класу та кількох обґрунтованих похідних (але не менше трьох) класів. Вибір відношень між класами (наслідування одиночне або множинне, відкрите або закрите) повинен обґрунтовуватися. Реалізувати віртуальне наслідування.
2. Визначити в абстрактний клас.
3. Визначити на базі кожного похідного класу методи, конструктори без параметрів, з параметрами, копіювання (базовий та похідні класи повинні містити не менше п'яти методів). Визначити в частині класів деструктори, а також віртуальні деструктори та продемонструвати відмінність їх роботи.
4. Кожен клас, що входить в ієрархію, що використовує відкрите наслідування, повинен містити принаймні одну віртуальну функцію. Функціональність зазначених функцій не повинна дублюватися.
5. Створити 2 типи об'єктів – статичні та динамічні (по 5). Передбачити усі можливі виключні ситуації роботи з динамічною пам'яттю (з демонстрацією роботи через try-catch та throw - не менше трьох).
6. Для видалення об'єкта з масивів передбачити методи Service\_Static та Service\_Dynamic. Видалення об'єкта даним методом здійснюється за певним критерієм (задається студентом самостійно, наприклад: колір, розмір, кількість, більше, менше, числові дані тощо). Передбачити усі можливі виключні ситуації роботи з динамічною пам'яттю (з демонстрацією роботи через try-catch та throw - не менше трьох).
7. Визначити в функцію Service, яка в якості параметра отримує об'єкт класу, і яка оперуючи визначеними раніше методами Read, Write. Вказані методи записують у файл та читають з файлу поля даних. Передбачити усі можливі виключні ситуації роботи з файлами (з демонстрацією роботи через try-catch та throw - не менше п'яти - не менше п'яти). **Ім'я створюваного файлу є параметром, що задається в командному рядку (робота програми здійснюється саме з виклику даного методу у функції main() ).**
8. Визначити класі статичний метод для перегляду створених об'єктів. Статичний метод перегляду об'єктів викликати не через об'єкт, а через клас.
9. Програма повинна надавати користувачу можливість вибору створення та знищення певного об'єкта (основні сутності) засобами меню.
10. Продемонструвати **поліморфні виклики усіх методів**.
11. Програма використовує три файли: заголовний h-файл із визначенням класу, сpp-файл із реалізацією класу, сpp-файл демонстраційної програми.

## Варіанти завдань.

1. Система **Університет**, кожен **Студент** може записуватись на **Факультет**, та складати іспити. **Викладачі** можуть переглядати списки записаних студентів, а також виставляти оцінки за іспити, якщо є оцінки менше за 3, то Викладач відраховує Студента.
2. Система **Прокат автомобілів**. Клієнт вибирає **Автомобіль** зі списку доступних, заповнює форму **Замовлення**, вказуючи паспортні дані, строк оренди. **Адміністратор** може відхилити **Заявку**, указавши причини відмови. При підтвердженні **Заявки** Клієнт оплачує **Замовлення**. Система виписує суму. У випадку ушкодження **Автомобіля** Клієнтом **Адміністратор** вносить відповідні позначки.
3. У кожному **Місті** є декілька **Поштових відділень**, які приймають та відсилають **Посилки**. Визначається їх вага, дата відправки та доставки, ціна. Клієнт відправляє посилку і сплачує кошти за її доставку.
4. На **Турнірі** реєструються **Футбольні команди**. Визначається час та місце проведення матчів. Про це повідомляється **Вболівальникам**, які можуть купувати квитки через інтернет та у касах. **Футболісти** відвідують тренування та відіграють матчі на турнірі, за що їм нараховується зарплата. Також, якщо команда виграє турнір, вона отримує певні кошти.
5. Система **Періодичні видання**. **Читач** може зробити **Заявку**, попередньо вибравши періодичні **Видання** зі списку. Система підраховує суму для оплати. **Читач** сплачує **заявку**. **Адміністратор** додає **Заявку** в «чорний список», якщо Клієнт не сплачує її в певний строк.
6. Система **Замовлення готелю**. Клієнт залишає **Заявку** на **Номер**, вказавши кількість місць у номері, клас апартаментів і час перебування. **Адміністратор** розглядає **Заявку**, підтверджує або відхиляє її. Результат переглядає Клієнт. У випадку підтвердження **Заявки** Клієнт оплачує послуги.
7. У **Перукарні Менеджер** приймає замовлення від Клієнтів, та формує черги до **Перукарів**. Перукар обслуговує клієнта, за що отримує кошти та надає каталог можливих зачісок. Клієнт записується, обирає стрижку з каталогу та оплачує стрижку.
8. **Туристична агенція** надає певний пакет послуг **Туристу**, який може замовити **Путівку**, оплатити її, та обрати певну **Країну** для туристичної подорожжі.
9. **Замовнику** потрібно розробити певний **Проект**, він замовляє його у **Менеджера**, який розподіляє роботу між **Робітниками** офісу.
10. **Університет** надає **Путівки** через **Профспілку**, куди можуть реєструватись **Студенти**. Вони можуть обирати місто для поїздки за певну оплату.
11. **Комп'ютер** налічує такі пристрої як **Монітор**, **Миша**. Вони підключаються до **Системного блоку**. Потрібно встановити драйвера і протестувати їх роботу.
12. **Відвідувачі** купують квитки на **Концерт** у касі. **Менеджер** розраховує час та місце концерту і повідомляє **Співака**.
13. Система **Факультатив**. **Викладач** оголошує запис на **Курс**. **Студент** записується на **Курс**, навчається і по закінченню **Викладач** виставляє **Оцінку**, яка зберігається в **Архіві**. **Студентів**, **Викладачів** і **Курсів** при навчанні може бути кілька.
14. Система **Платежі**. Клієнт має **Рахунок** у банку і **Кредитну Карту (КК)**. Клієнт може оплатити **Замовлення**, зробити платіж на інший **Рахунок**, заблокувати **КК** і анулювати **Рахунок**. **Адміністратор** може заблокувати **КК** за перевищення кредиту.
15. Система **Житлово-комунальні послуги**. **Квартиронаймач** відправляє **Заявку**, у якій вказує вид робіт, масштаб і бажаний час виконання. **Диспетчер** формує відповідну **Бригаду** і реєструє її в **Плані робіт**. **Диспетчер** може відхилити **Заявку** у випадку зайнятості всіх **Бригад**.
16. Система **Лікарня**. **Пацієнтові** призначається лікар. **Лікар** може зробити лікарські призначення **Пацієнтові** (процедури, ліки, операції). **Медсестра** або інший **Лікар** виконують призначення. **Пацієнт** може бути виписаний з **Лікарні** по закінченню лікування, при порушенні режиму або при інших обставинах.
17. Система **Вступні іспити**. **Абітурієнт** реєструється на **Факультет**, складає **Іспити**. **Викладач** виставляє **Оцінку**. Система підраховує середній бал і визначає **Абітурієнтів**, зарахованих у навчальний заклад.
18. Система **Бібліотека**. **Читач** оформляє **Замовлення** на **Книгу**. Система здійснює пошук у **Каталозі**. **Бібліотекар** видає **Читачеві Книгу** на абонемент або в читальний зал. При неповерненні **Книги** **Читачем** він може бути занесений **Адміністратором** в «чорний список».