**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**



### ЗВІТ

До лабораторної роботи № 5

**На тему:** *“ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ З ВИКОРИСТАННЯМ UML”*

**З дисципліни:** *“Вступ до інженерії програмного забезпечення”*

**Лектор:**

доцент каф. ПЗ

Левус Є. В.

**Виконав:**

ст. гр. ПЗ-15

Шпортко Т. О.

**Прийняв:**

асистент каф. ПЗ

Самбір А. А.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2023 р.

∑= \_\_\_\_ .

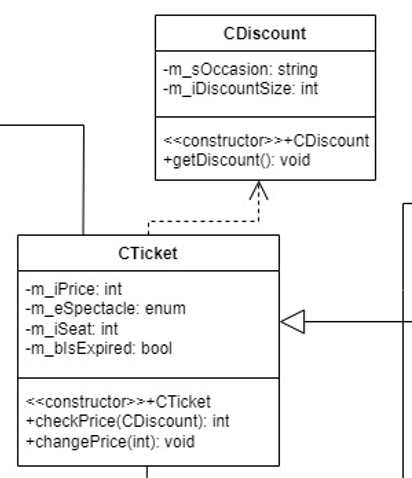
Львів – 2023

**Тема роботи:** Опис предметної області з використанням UML.

**Мета роботи:** Навчитися створювати об’єктну модель програмної системи.

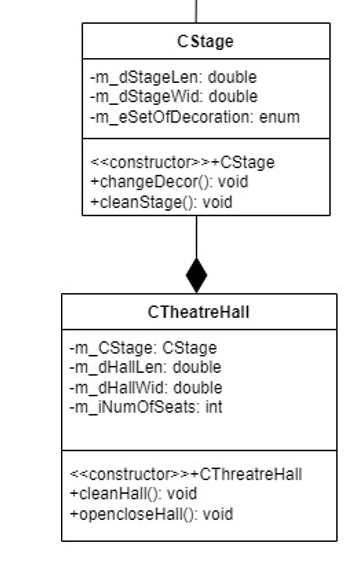
**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Як відображено у класі на рівні реалізації:



Відношення «джерело змін – змінююча сутність»

(як параметр методу класу)



Відношення «частина-ціле»

(як атрибут класу)

**ПОСТАНОВКА ЗАВДАННЯ**

Індивідуальний варіант: 15. Інформаційна система «Театр».

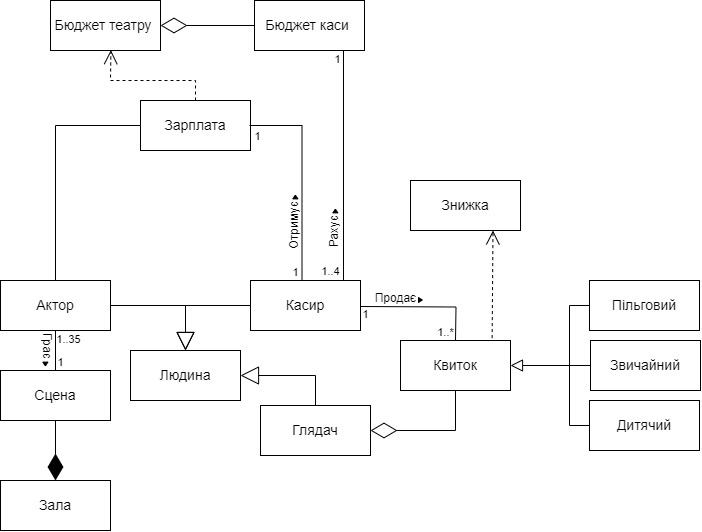
Згідно індивідуального варіанту провести аналіз предметної області. Для виконання завдання:

1. Скласти словник предметної області.
2. Побудувати UML-діаграму класів на концептуальному рівні засобами програми Draw.io. Зобразити коментарі на схемі. Вказати відношення між сутностями (узагальнення, звичайна асоціація, агрегація, композиція) із обов’язковим зазначенням їх характеристик (кратність, назва асоціації, і т.п.).
3. Побудувати UML-діаграму конкретних класів на рівні реалізації засобами програми Draw.io. Чітко вказати усі поля та методи класів з відповідними модифікаторами доступу, а також усі необідні відношення між класами.
4. Оформити звіт.

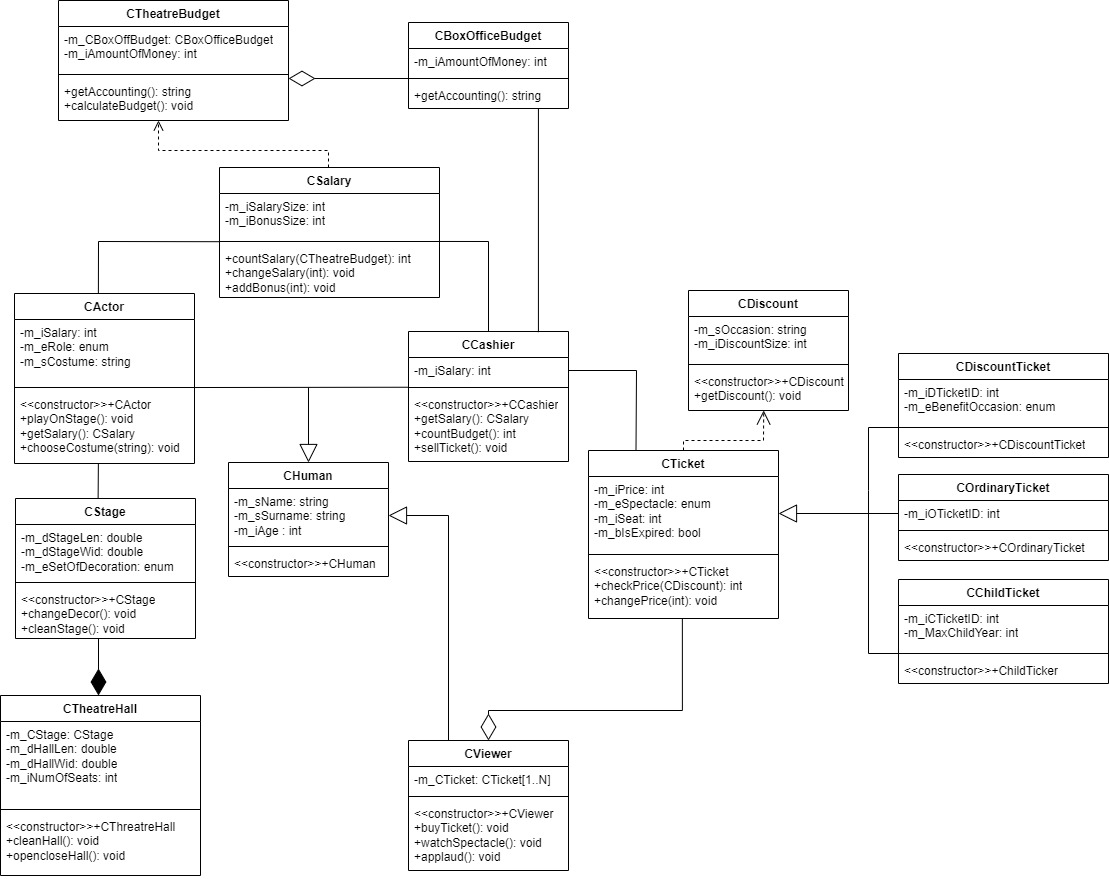
**ХІД РОБОТИ**

**Словник предметної області**

1. **Actor –** сутність актор, має властивості (*містить атрибути*) *salary* (зарплатня), *role* (роль), *costume* (костюм) і може грати на сцені (*метод playOnStage()*), отримати зарплатню (*метод getSalary()*) та змінити костюм (*метод chooseCostume()*). Зв’язаний з сутностями (*класами*) **Salary** та **Stage** (відношення асоціації) є спеціалізованою сутністю (*похідним класом*) сутності **Human**.
2. **BoxOfficeBudget –** сутність бюджет каси, має властивості (*містить атрибут*) *amountofmoney* (кількість грошей), можна отримати облік (*метод getAccounting()*). Зв’язаний з сутностями (*класами*) **TheatreBudget** (відношення агрегації, є частиною), **Cashier** (відношення асоціації).
3. **Cashier –** сутність касир, має властивості (*містить атрибути*) *salary* (зарплатня) і може отримати зарплатню (*getSalary()*), підрахувати бюджет каси (*countBudget()*) та продати квиток (*метод sellTicket()*). Зв’язаний з сутностями (*класами*) **Salary**, **Ticket** та **BoxOfficeBudget** (відношення асоціації) є спеціалізованою сутністю (*похідним класом*) сутності **Human**.
4. **ChildTicket –** сутність дитячий квиток, має властивості (*містить атрибути*) *cticketID* (номер дитячого квитка), *maxchildyear* (максимальний вік людини для покупки дитячого квитка). Є спеціалізованою сутністю (*похідним класом*) сутності **Ticket**.
5. **Discount –** сутність знижка, має властивості (*містить атрибути*) *occasion* (причина отримання знижки), *discountsize* (розмір знижки), її можна отримати (*метод getDiscount()*). Зв’язаний з сутністю (*класом*) **Ticket** (відношення залежності, є джерелом змін).
6. **DiscountTicket –** сутність пільговий квиток, має властивості (*містить атрибути*) *dticketID* (номер пільгового квитка), *benefitoccasion* (пільга). Є спеціалізованою сутністю (*похідним класом*) сутності **Ticket**.
7. **Human –** сутність людина (*базовий клас*), має властивості (містить атрибути) *name* (ім’я), *surname* (прізвище) та *age* (вік). Спеціалізований сутностями **Cashier**, **Viewer** і **Actor** (*дочірніми класами*).
8. **OrdinaryTicket –** сутність звичайний квиток, має властивості (*містить атрибути*) *oticketID* (номер звичайного квитка). Є спеціалізованою сутністю (*похідним класом*) сутності **Ticket**.
9. **Salary – сутність заробітна плата,** має властивості (*містить атрибути*) *salarysize* (розмір зарплатні), *bonussize* (розмір бонусу) і можна підрахувати зарплатню (*метод countSalary()*), змінити зарплатню (*метод changeSalary()*) та додати бонус (*метод addBonus()*). Зв’язаний з сутностями (*класами*) **TheatreBudget** (відношення залежності, джерело змін – змінююча сутність), **Actor** та **Cashier** (відношення асоціації).
10. **Stage –** сутність сцена, має властивості (*містить атрибути*) *stagelen* (довжина сцени), *stagewid* (ширина сцени), *setofdecorations* (набір декорацій) і може бути прибрана (*метод cleanStage()*), також можна змінити набір декорацій (*метод changeDecor()*). Зв’язаний з сутностями (*класами*) **TheatreHall** (відношення композиції, є частиною) та **Actor** (відношення асоціації).
11. **TheatreBudget –** сутність бюджет театру, має властивості (*містить атрибути*) *BoxOffBudget* (бюджет каси), *amountofmoney* (кількість грошей) і може бути підрахований (*метод calculateBudget()*), також можна отримати облік (*метод getAccounting()*). Зв’язаний з сутностями (*класами*) **Salary** (відношення залежності, є джерелом змін) та **BoxOfficeBudget** (відношення агрегації, є цілим).
12. **TheatreHall –** сутність зала, має властивості (*містить атрибути*) *Stage* (сцена), *halllen* (довжина зали), *hallwid* (ширина зали), *numofseats* (кількість сидінь) і може бути прибрана (*метод cleanHall()*), закрита/відкрита (*opencloseHall()*). Зв’язаний з сутністю (класом) **Stage** (відношення композиції, є цілим).
13. **Ticket –** сутність квиток (*базовий клас*), має властивості (*містить атрибути*) *price* (ціна), *spectacle* (вистава), *seat* (номер місця/сидіння), *isexpired* (чи дійсний) і можна перевірити ціну (*метод checkPrice()*) та змінити її (*метод changePrice()*). Зв’язаний з сутностями (*класами*) **Discount** (відношення залежності, джерело змін – змінююча сутність), **Cashier** (відношення асоціації), **Viewer** (відношення агрегації, є частиною) і спеціалізований сутностями **DiscountTicket**, **OrdinaryTicket** та **ChildTicket**.
14. **Viewer –** сутність глядач, має властивості (*містить атрибути*) *Ticket[1..N]* (квиток) і може придбати квиток (*метод buyTicket()*), переглянути виставу (*метод watchSpectacle()*) та аплодувати (*метод applaud()*). Зв’язаний з сутністю (*класом*) **Ticket** (відношення агрегації, є цілим) та є спеціалізованою сутністю (*похідним класом*) сутності **Human**.

**UML-діаграма на концептуальному рівні**

**UML-діаграма на рівні реалізації**

****

**ВИСНОВКИ**

Виконуючи цю лабораторну роботу, я провів аналіз предметної області, навчився створювати UML-діаграми на концептуальному рівні і рівні реалізації за допомогою раніше вивчених сутностей мови UML та взаємозв’язків між ними, попередньо склавши словник предметної області згідно з індивідуальним варіантом («Театр»).