

ЛАБОРАТОРНА РОБОТА №1.

ОСНОВИ РОЗРОБКИ КЛАСІВ МОВОЮ C++

Виконати індивідуальне завдання відповідно до номеру в списку групи:

1.1 Завдання на лабораторну роботу

Кожний побудований клас повинен реалізовувати:

1. конструктор за замовченням
2. деструктор
3. конструктори с параметрами
4. перевантажену операцію присвоювання
5. перевантажену операцію виведення в потік
6. перевантажену операцію введення з потоку
7. перевантажену арифметичну операцію
8. перевантажену логічну операцію
9. перевантажену операцію приведення до типу (наприклад для отримання середнього значення);
10. додаткові методи, якщо це необхідно для реалізації завдання

Реалізація елементів-даних не регламентується, потрібно тільки, щоб клас повністю реалізовував заданий інтерфейс. Рекомендується, де це необхідно, розробляти допоміжні класи. Деякі необхідні операції можуть бути операціями цих допоміжних класів.

1.2 Варіанти завдань

ВАРІАНТ 1

Розробити клас, що представляє студента. Студент характеризується ім'ям, прізвищем, групою і набором іспитів, які він складав. Іспит

характеризується назвою предмета, оцінкою студента по ньому і датою складання (рік, семестр). Група характеризується курсом і факультетом.

Розробити методи які дозволяють:

1. дізнатися повне ім'я студента (ім'я + прізвище) і його курс
2. дізнатися, чи вчиться він на заданому факультеті
3. дізнатися найвищу оцінку серед усіх іспитів з даного предмету
4. додати йому оцінку за іспит (назва передається як параметр)
5. видалити для нього оцінку за іспит (назва передається як параметр)
6. дізнатися число іспитів, які він склав зі вказаною оцінкою
7. дізнатися його середній бал за вказаний семестр

ВАРІАНТ 2

Розробити клас, що представляє гуртожиток. Гуртожиток характеризується вулицею, номером будинку, факультетом і набором кімнат. Кімната характеризується номером, місткістю і числом зайнятих місць. Факультет характеризується інститутом, назвою і числом студентів.

Розробити методи які дозволяють:

1. дізнатися повну адресу гуртожитку і до якого інституту він відноситься
2. дізнатися середню заселеність гуртожитку
3. додати для нього кімнату (відкрити її для заселення)
4. зайняти кімнату (з перевіркою)
5. звільнити кімнату
6. дізнатися число вільних кімнат (повністю/частково)
7. дізнатися, який відсоток студентів інституту живе в гуртожитку

ВАРІАНТ 3

Розробити клас, що представляє факультет. Факультет

характеризується повною і короткою назвою, інститутом, до якого він належить, і списком навчальних груп. Інститут характеризується назвою та адресою. Група характеризується назвою, числом студентів і середнім балом за підсумками останньої сесії.

Розробити методи які дозволяють:

1. дізнатися повну і коротку назву факультету та його інститут
2. дізнатися, чи відноситься ця назва до даного інституту
3. дізнатися кількість студентів, у яких бал вище середнього
4. додати групу (початок першого курсу)
5. видалити групу (після диплома)
6. дізнатися число студентів даної групи
7. дізнатися групу з найвищим середнім балом

ВАРІАНТ 4

Розробити клас, що представляє викладача. Викладач характеризується ім'ям, прізвищем, кафедрою, списком предметів, які він викладає. Кафедра відноситься до факультету і має ім'я. Предмет характеризується назвою, кількістю годин і середнім балом, виставленим викладачем на останній сесії по всіх групах.

Розробити методи які дозволяють:

1. дізнатися ім'я і прізвище викладача та його кафедру
2. дізнатися, чи працює він на даному факультеті
3. дізнатися середню кількість годин для всіх предметів
4. додати предмет (внесли до навчального плану)
5. видалити предмет (виключили з навчального плану)
6. знайти найнижчий середній бал з усіх предметів
7. дізнатися, чи існує предмет із заданою кількістю годин

ВАРІАНТ 5

Розробити клас, що представляє методичний посібник. Посібник характеризується автором, назвою, предметом і списком видач примірників на руки. Видача характеризується датою і ім'ям студента. Автор характеризується ім'ям, прізвищем і назвою кафедри.

Розробити методи, які дозволяють:

1. дізнатися ім'я і прізвище автора посібника
2. дізнатися назву, предмет посібника та кафедру
3. дізнатися загальну кількість виданих посібників
4. видати посібник на руки
5. отримати посібник назад
6. знайти студентів які отримали посібник декілька раз.
7. дізнатися число видач для заданої дати

ВАРІАНТ 6

Розробити клас, що представляє книгу в бібліотеці. Книга характеризується списком авторів, назвою і темою. Тема характеризується кодом теми і назвою теми. Автор характеризується ім'ям, прізвищем, кращою темою і числом написаних книг.

Розробити методи які дозволяють:

1. дізнатися назву і тему книги
2. дізнатися, чи написав даний автор цю книгу
3. дізнатися середнє число написаних книг для всіх авторів
4. додати автора до книги
5. видалити автора з книги з перевіркою за ім'ям та прізвищем
6. знайти автора з максимальним числом написаних книг
7. знайти число авторів, чия тема краще збігається з темою книги

ВАРІАНТ 7

Розробити клас, що представляє студентську групу. Група характеризується факультетом, назвою і списком студентів. Студент характеризується ім'ям, прізвищем і середнім балом за останню сесію. Факультет характеризується назвою і профілем діяльності.

Розробити методи, які дозволяють:

1. дізнатися назву і факультет групи
2. дізнатися, чи відноситься дана група до даного факультету
3. дізнатися середній бал, розрахований за всіма студентами з даним ім'ям
4. прийняти студента в групу
5. виключити студента з групи
6. знайти число студентів з балом вище середнього
7. знайти середній бал зазначеного студента

ВАРІАНТ 8

Розробити клас, що представляє університет. Університет характеризується назвою, вулицею, номером будинку, списком факультетів і ректором. Ректор характеризується ім'ям, прізвищем і вченим ступенем. Факультет характеризується назвою, профілем і числом студентів.

Розробити методи, які дозволяють:

1. дізнатися повну адресу університету
2. дізнатися ім'я, прізвище і вчений ступінь ректора
3. дізнатися число студентів даного факультету
4. відкрити новий факультет
5. закрити факультет
6. знайти середнє число студентів за всіма факультетами
7. знайти загальне число студентів всіх факультетів даного профілю

ВАРІАНТ 9

Розробити клас, що представляє лекційний курс. Лекційний курс характеризується назвою, викладачем, який його читає, групою і списком проведених пар. Пара характеризується датою і числом студентів, які відвідали її. Група характеризується назвою та кількістю студентів.

Розробити методи, які дозволяють:

1. дізнатися назву курсу та ім'я викладача
2. дізнатися число студентів групи, якій він читається
3. визначити середній відсоток відвідуваності за всіма парами
4. провести пару
5. викреслити проведену пару
6. знайти число пар, на які прийшло задане число студентів
7. дізнатися, чи була стовідсоткова відвідуваність, якщо була, то коли в перший раз.

ВАРІАНТ 10

Розробити клас, що представляє навчальний корпус інституту. Навчальний корпус відноситься до інституту і характеризується назвою та списком аудиторій. Інститут характеризується адресою та назвою. Аудиторія характеризується номером, місткістю і зайнятістю (зайнята/не зайнята).

Розробити методи які дозволяють:

1. дізнатися назву інституту, назву корпусу і адресу
2. дізнатися, чи відноситься даний корпус до даного інституту
3. визначити середню місткість вільних аудиторій
4. додати аудиторію (відкрити її для занять)
5. видалити аудиторію (закрити на ремонт)
6. дізнатися, чи зайнята дана аудиторія

7. знайти сумарну місткість корпусу

ВАРІАНТ 11

Розробити клас, що представляє магазин. Магазин відноситься до міста і характеризується назвою та списком продуктів. Місто характеризується назвою, областю, в якій воно розташоване, та кількістю населення. Продукт характеризується назвою, кількістю, масою і ціною.

Розробити методи, які дозволяють:

1. дізнатися назву міста, назву області
2. дізнатися, чи відноситься даний магазин до вказаної області
3. визначити середню масу усіх продуктів
4. додати новий продукт
5. видалити продукт (зняти з продажі)
6. дізнатися, чи є у наявності вказаний продукт
7. знайти сумарну кількість усіх продуктів

ВАРІАНТ 12

Розробити клас, що представляє бібліотеку. Бібліотека характеризується назвою, адресою, спеціалізацією та списком книг. Адреса характеризується містом, вулицею, номером будинку. Книга характеризується назвою, роком видання, кількістю у бібліотеці, кількістю виданих книг.

Розробити методи, які дозволяють:

1. дізнатися назву та адресу бібліотеки
2. перевірити наявність книги за назвою
3. дізнатися середнє число виданих книг
4. додати книгу до бібліотеки
5. видалити книгу з бібліотеки

6. видати книгу на руки (збільшити кількість виданих книг)
7. підрахувати процент книжок, які повністю видані

ВАРІАНТ 13

Розробити клас, що представляє аеропорт. Аеропорт характеризується назвою, містом і набором літаків. Літак характеризується номером, місткістю і числом зайнятих місць. Місто характеризується назвою, областю, в якій воно розташоване, та кількістю населення.

Розробити методи, які дозволяють:

1. дізнатися назву міста, назву області
2. дізнатися середню зайнятість літаків
3. додати до аеропорту новий літак (відкрити його для посадки)
4. зайняти літак (з перевіркою)
5. списати літак
6. дізнатися число вільних місць (повністю/частково)
7. дізнатися, який відсоток літаків повністю заповнені

ВАРІАНТ 14

Розробити клас, що представляє залізничний вокзал. Вокзал характеризується назвою, адресою, і набором поїздів. Поїзд характеризується номером, типом (швидкий/звичайний), місткістю і числом зайнятих місць. Адреса характеризується містом, вулицею, номером будинку.

Розробити методи, які дозволяють:

1. дізнатися повну адресу вокзалу
2. дізнатися середню зайнятість поїздів
3. додати до вокзалу новий поїзд (відкрити його для посадки)
4. зайняти поїзд (з перевіркою)
5. списати (видалити) поїзд

6. дізнатися число вільних місць (повністю/частково)
7. дізнатися відсоток поїздів, які повністю заповнені

ВАРІАНТ 15

Розробити клас, що представляє поїзд. Поїзд характеризується назвою, номером, вокзалом, до якого він приписаний, і набором вагонів. Вокзал характеризується назвою, назвою міста, в якому він розташований. Вагон характеризується номером та кількістю місць.

Розробити методи, які дозволяють:

1. дізнатися назву вокзалу, до якого приписаний поїзд
2. дізнатися середню кількість вільних місць у вагонах
3. додати до поїзда новий вагон (відкрити його для посадки)
4. зайняти місце у конкретному вагоні (з перевіркою)
5. звільнити місце у конкретному вагоні
6. дізнатися число вільних місць у поїзді/конкретному вагоні
7. дізнатися відсоток вагонів, які повністю зайняті

ВАРІАНТ 16

Розробити клас, що представляє гру шахи. Шахи характеризуються назвою, виробником і набором шахових фігур. Виробник характеризується назвою, адресою розташування. Адреса характеризується містом, вулицею, номером будинку. Шахова фігура характеризується назвою (пішак, кінь, тура тощо), кольором (чорний/білий), координатами розташування на дошці (наприклад B4).

Розробити методи, які дозволяють:

1. дізнатися назву виробника шахів та його розташування
2. дізнатися загальну кількість фігур на дошці за типом (тип передається у якості параметра)

3. додати нову фігуру на дошку з перевіркою вільності клітинки за координатами

4. пересунути фігуру на вказану позицію з перевіркою зайнятості клітинки

5. забрати (видалити) фігуру з дошки за координатами

6. дізнатися число вільних клітинок на дошці

7. дізнатися, який відсоток фігур на дошці

ВАРІАНТ 17

Розробити клас, що представляє колоду карт. Колода характеризується назвою, виробником, набором карт, загальною кількістю карт. Виробник характеризується назвою, адресою розташування. Адреса характеризується містом, вулицею, номером будинку. Карта характеризується назвою (туз, король, дама тощо), мастю (чирва, піки тощо).

Розробити методи які дозволяють:

1. дізнатися назву виробника колоди та його розташування

2. дізнатися загальну кількість карт у колоді за мастю (масть передається у якості параметра)

3. додати нову карту у колоду з перевіркою, що такої карти ще немає в колоді (карта повинна бути унікальна за назвою та мастю)

4. підрахувати кількість карт за назвою (без урахування масті, наприклад, кількість тузів)

5. забрати (видалити) карту з колоди за назвою та мастю

6. дізнатися число відсутніх карт

7. дізнатися, який відсоток карт у колоді

ВАРІАНТ 18

Розробити клас, що представляє охорону замку. Охорона характеризується назвою, замком, до якого вона належить, і загonom воїнів.

Замок характеризується назвою, типом замку, рівнем розвитку. Воїн характеризується назвою, типом (лучники, дракони тощо), рівнем життя, рівнем захисту, силою удару.

Розробити методи які дозволяють:

1. дізнатися повну інформацію про замок, до якого відноситься охорона
2. дізнатися загальну кількість воїнів конкретного типу (тип передається у якості параметра)
3. додати нового воїна у загін
4. підрахувати середню силу удару воїнів
5. звільнити (видалити) воїна з загону за назвою
6. дізнатися число воїнів зі 100 процентним рівнем життя
7. дізнатися, який загальний відсоток рівня захисту в загоні

ВАРІАНТ 19

Розробити клас, що представляє готель. Готель характеризується адресою, набором кімнат та типом готелю. Адреса характеризується назвою міста, вулиці, номером будинку. Кімната характеризується номером, рівнем (люкс/стандарт), місткістю і числом зайнятих місць. Тип готелю характеризується наявністю конференц-залу, ресторану.

Розробити методи, які дозволяють:

1. дізнатися повну адресу готелю і до якого типу він відноситься
2. дізнатися середню заселеність готелю
3. додати до готелю кімнату (відкрити її для заселення)
4. зайняти кімнату (з перевіркою)
5. звільнити кімнату
6. дізнатися число вільних кімнат (повністю/частково)
7. дізнатися, який відсоток кімнат повністю зайняті

ВАРІАНТ 20

Розробити клас, що представляє книжку. Книжка характеризується назвою, жанром, списком авторів, загальним тиражем, роком видання та кількістю проданих книжок. Жанр характеризується назвою, віковою аудиторією. Автор характеризується ім'ям, прізвищем, датою народження.

Розробити методи, які дозволяють:

1. дізнатися назву та жанр книжки
2. дізнатися рік видання книжки
3. дізнатися середній вік авторів
4. додати автора до книжки
5. видалити автора
6. знайти найстаршого автора
7. дізнатися процент проданих книжок відповідно до тиражу

ВАРІАНТ 21

Розробити клас, що представляє компанію. Компанія характеризується назвою, адресою, видом діяльності, списком відділів, роком заснування. Адреса характеризується назвою міста, вулиці, номером будинку. Відділ характеризується назвою, спеціалізацією, кількістю працівників, процентом завантаження.

Розробити методи, які дозволяють:

1. дізнатися назву та адресу компанії
2. дізнатися рік заснування компанії
3. дізнатися середнє завантаження компанії
4. додати новий відділ з перевіркою по назві
5. видалити відділ
6. знайти найбільш завантажений відділ
7. дізнатися середню кількість працівників у відділі

ВАРІАНТ 22

Розробити клас, що представляє відділ. Відділ характеризується назвою, компанією, до якої відноситься, спеціалізацією діяльності, списком працівників. Компанія характеризується назвою та адресою. Адреса характеризується назвою міста, вулиці, номером будинку. Працівник характеризується ім'ям, спеціалізацією, датою народження, кількістю задач, які виконує.

Розробити методи, які дозволяють:

1. дізнатися назву відділу та компанії, до якої він відноситься
2. дізнатися адресу розташування
3. дізнатися середню кількість задач, які виконує працівник
4. додати нового працівника
5. звільнити працівника
6. знайти працівника, у якого найбільше задач
7. дізнатися середній вік працівника у відділі

ВАРІАНТ 23

Розробити клас, що представляє ботанічний сад. Ботанічний сад характеризується назвою, адресою розташування, роком заснування, списком рослин. Адреса характеризується назвою міста, вулиці, номером будинку. Рослина характеризується типом, роком, в якому була посаджена, GPS координатами розташування.

Розробити, методи які дозволяють:

1. дізнатися назву ботанічного саду
2. дізнатися адресу розташування
3. дізнатися середній вік рослин у ботанічному саду
4. додати нову рослину, з перевіркою по координатам
5. видалити рослину

6. знайти найстаршу рослину
7. дізнатися тип рослин, яких найбільше в ботанічному саду

ВАРІАНТ 24

Розробити клас, що представляє ферму. Ферма характеризується назвою, адресою розташування, роком заснування, списком тварин. Адреса характеризується назвою міста, вулиці, номером будинку. Тварина характеризується видом, роком народження, вагою.

Розробити, методи які дозволяють:

1. дізнатися назву ферми
2. дізнатися адресу розташування
3. дізнатися середній вік тварин для заданого виду
4. додати нову тварину
5. видалити тварину
6. знайти найважчу тварину
7. дізнатися найважчий тип тварин за середньою вагою

ВАРІАНТ 25

Розробити клас, що представляє лікарню. Лікарня характеризується назвою, адресою розташування, спеціалізацією, списком пацієнтів. Адреса характеризується назвою міста, вулиці, номером будинку. Пацієнт характеризується ім'ям, прізвищем, роком народження, датою поступлення в лікарню, назвою хвороби, яку лікує.

Розробити методи, які дозволяють:

1. дізнатися назву лікарні
2. дізнатися адресу розташування
3. дізнатися середній вік пацієнта по заданій хворобі
4. додати нового пацієнта

5. виписати пацієнта
6. знайти пацієнта, який найдовше знаходиться в лікарні
7. дізнатися середній час перебування пацієнта в лікарні відповідно до хвороби

ВАРІАНТ 26

Розробити клас, що представляє спеціальність. Спеціальність характеризується назвою, факультетом, спеціалізацією, списком предметів, які вивчають студенти. Факультет характеризується назвою, назвою університету, до якого відноситься. Предмет характеризується назвою, номером семестру, загальною кількістю годин, кількістю лекційних годин, кількістю практичних годин.

Розробити методи, які дозволяють:

1. дізнатися назву спеціалізації та факультет, до якого відноситься
2. дізнатися спеціалізацію спеціальності
3. дізнатися загальну кількість годин по усім предметам
4. додати новий предмет з перевіркою по назві
5. видалити предмет
6. знайти предмет з найбільшою кількістю лекційних годин
7. дізнатися середнє відношення лекційних годин до практичних

ВАРІАНТ 27

Розробити клас, що представляє кінотеатр. Кінотеатр характеризується назвою, адресою розташування, списком фільмів, які демонструються, роком заснування. Адреса характеризується назвою міста, вулиці, номером будинку. Фільм характеризується назвою, роком створення, назвою жанру, віковим обмеженням, тривалістю фільму у хвилинах.

Розробити методи, які дозволяють:

1. дізнатися назву кінотеатру
2. дізнатися адресу розташування
3. дізнатися процентну кількість фільмів, які відносяться до дитячої категорії
4. додати новий фільм
5. видалити фільм
6. знайти найдовший фільм за часом
7. дізнатися середню тривалість фільму

1.3 Контрольний приклад

Розробити клас, що представляє фільм. Фільм характеризується назвою, країною, де був створений, жанром, списком акторів, які знімалися у фільмі, віковою аудиторією, тривалістю фільму. Актор характеризується ім'ям, прізвищем, роком народження, роком, у якому перший раз знявся у фільмі.

Розробити методи, які дозволяють:

1. дізнатися назву фільму
2. дізнатися назву жанру
3. дізнатися середній вік актора
4. додати нового актора
5. звільнити актора
6. знайти найстаршого актора
7. дізнатися середній стаж актора у фільмі

Загальні завдання:

1. конструктор за замовчуванням
2. деструктор
3. конструктори с параметрами
4. конструктор копіювання
5. перевантажену операцію присвоювання

6. перевантажену операцію виведення в потік
7. перевантажену операцію введення з потоку
8. перевантажену арифметичну операцію
9. перевантажену логічну операцію
10. перевантажену операцію приведення до типу (наприклад, для отримання середнього значення)

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <string>
#include <ctime>
using namespace std;
enum Tgenre { ACTION, WESTERN, MYSTERY, DRAMA, COMEDY, ROMANCE, ADVENTURE, THRILLER,
HISTORY, HORROR, FANTASY, NUMBER_GENRE};
const char *listNameGetre[NUMBER_GENRE] = { "ACTION", "WESTERN", "MYSTERY", "DRAMA",
"COMEDY", "ROMANCE", "ADVENTURE", "THRILLER", "HISTORY", "HORROR", "FANTASY"};
//note example print name on console: cout << listNameGetre[WESTERN] << endl;
struct Tactor
{
    string firstName;
    string secondName;
    int yearBirth;
    int yearFirstMove;
};
ostream& operator <<(ostream& out, const Tactor& actor);
istream& operator >>(istream& in, Tactor& actor);

class TMovie
{
private:
    string nameMovie;
    string country;
    Tgenre genre;
    Tactor *actors; //it is dynamic array
    int n; //number actors
    int ageaudience;
    int lengthOfFilm;

public:
    TMovie() ;
    TMovie(const TMovie&obj);
    TMovie(const string& nameMovie_, const string& country_, const Tgenre& genre_, int
ageaudience_, int lengthOfFilm_);
    ~TMovie();
    const string& getNameMovie()const;
    const string& getCountry()const;
    const char*      getNameGenre()const;
    int              getAgeaudience()const;
    int              getLengthOfFilm()const;
    int              getNumberActors()const;
    void              outputActors(ostream& out)const;
    void              setNameMovie(const string& nameMovie_);
    void              setCountry(const string& country_);
    void              setNameGenre(const Tgenre& genre_);
    void              setAgeaudience(int ageaudience_);
    void              setLengthOfFilm(int lengthOfFilm_);
};
```

```

        int                calcAverageAgeActor()const;
        bool               isExistdActor(string firstName_, string secondName_)const;
        bool               addActor(string firstName_, string secondName_, int
yearBirth_, int yearFirstMove_);
        bool               addActor(const TActor& actor);
        bool               deleteActor(string firstName_, string secondName_);
        TActor*            findOldestActor()const;
        int                calcAverageActorsTime()const;

        TMovie& operator =(const TMovie& obj);
        TMovie& operator +=(const TActor& actor);
        bool operator <(const TMovie& arg2) const;
        operator int()const;
};

ostream& operator <<(ostream& out, const TMovie& actor);
istream& operator >>(istream& in, TMovie& actor);

int main()
{
    time_t sysdate = time(NULL);
    cout << "Current Datetime:" << asctime(localtime(&sysdate)) << endl;
    TMovie chobit;
    cin >> chobit;
    cout << "Please add new actor:" << endl;
    TActor newActor;
    cin >> newActor;
    chobit+=newActor;
    cout << "Prepared info about film:\n";
    cout << chobit;
    cout << endl << endl;
    cout << "Average Actors Time: " << chobit.calcAverageActorsTime() << endl;
    cout << "Average age actor: " << chobit.calcAverageAgeActor() << endl;
    TActor *odestdActor = chobit.findOldestActor ();
    if (odestdActor) cout << "Oldestactor is :\n" << *odestdActor << endl;
    string firstName;
    string secondName;
    cout << "Please input name actor which need delete:\n";
    cout << "\tinput first name: ";
    getline(cin, firstName);
    cout << "\tinput second name: ";
    getline(cin, secondName);
    bool result = chobit.deleteActor(firstName, secondName);
    if (!result) cout << "Warning: Actor not found " << endl;
    cout << "Final info about film:\n";
    cout << chobit << endl;

    system("pause");
    return 0;
}

ostream& operator <<(ostream& out, const TActor& actor)
{
    out << "\t" << actor.firstName
        << "\t" << actor.secondName
        << "\t" << actor.yearBirth
        << "\t" << actor.yearFirstMove;
    return out;
}

istream& operator >>(istream& in, TActor& actor)
{
    cout << "Please input first name: ";
    getline(in, actor.firstName);
    cout << "Please input second name: ";

```

```

        getline(in, actor.secondName);
        cout << "Please input year of birth: ";
        string strTmp;
        getline(in, strTmp);
        actor.yearBirth = atoi(strTmp.c_str());
        cout << "Please input the year the first time starred in the film ";
        getline(in, strTmp);
        actor.yearFirstMove = atoi(strTmp.c_str());
        return in;
    }

    TMovie::TMovie():n(0), actors(NULL), nameMovie("New"), country("Ukrain"), ageaudience(0),
    lengthOfFilm(0), genre(ACTION)
    {
    }
    TMovie::TMovie(const TMovie&obj)
    {
        nameMovie = obj.nameMovie;
        country = obj.country;
        genre = obj.genre;
        ageaudience = obj.ageaudience;
        lengthOfFilm = obj.lengthOfFilm;
        n = obj.n;
        actors = new TActor[n];
        for (int i = 0; i < n; ++i)
        {
            actors[i] = obj.actors[i];
        }
    }

    TMovie::TMovie(const string& nameMovie_, const string& country_, const Tgenre& genre_,
    int ageaudience_, int lengthOfFilm_)
        :n(0), actors(NULL), nameMovie(nameMovie_), country(country_),
    ageaudience(ageaudience_), lengthOfFilm(lengthOfFilm_), genre(genre_)
    {
    }
    TMovie::~TMovie()
    {
        if (actors) delete[] actors;
        actors = NULL;
        n = 0;
    }
    const string& TMovie::getNameMovie()const
    {
        return nameMovie;
    }
    const string& TMovie::getCountry()const { return country;}
    const char* TMovie::getNameGenre()const { return listNameGetre[genre]; }
    int TMovie::getAgeaudience()const { return ageaudience; }
    int TMovie::getLengthOfFilm()const { return lengthOfFilm;}
    int TMovie::getNumberActors()const { return n; }
    void TMovie::outputActors(ostream& out)const
    {
        for (int i = 0; i < n; ++i)
            out << actors[i] << endl;
    }

    void TMovie::setNameMovie(const string& nameMovie_) { nameMovie = nameMovie_; }
    void TMovie::setCountry(const string& country_) { country = country_; }
    void TMovie::setNameGenre(const Tgenre& genre_) { genre = genre_; }
    void TMovie::setAgeaudience(int ageaudience_) { ageaudience = ageaudience_; }
    void TMovie::setLengthOfFilm(int lengthOfFilm_) { lengthOfFilm = lengthOfFilm_; }

    int TMovie::calcAverageAgeActor()const

```

```

{
    int result = 0;
    time_t sysdate = time(NULL);
    int curYear = 1900+localtime(&sysdate)->tm_year;
    for (int i = 0; i < n; ++i)
        result += (curYear - actors[i].yearBirth);
    result /= n;
    return result;
}
bool TMovie::isExistdActor(string firstName_, string secondName_)const
{
    for (int i = 0; i < n; ++i)
    {
        if(actors[i].firstName == firstName_ && actors[i].secondName ==
secondName_)
        {
            return true;
        }
    }
    return false;
}
bool TMovie::addActor(string firstName_, string secondName_, int
yearBirth_, int yearFirstMove_)
{
    TActor newActor = { firstName_, secondName_, yearBirth_, yearFirstMove_ };
    return addActor(newActor);
}
bool TMovie::addActor(const TActor& actor)
{
    if (isExistdActor(actor.firstName, actor.secondName)) return false;
    TActor * mas = new TActor[n + 1]; //выделяем новую область памяти
    for (int i = 0; i < n; ++i)
        mas[i] = actors[i]; //переносим массив в новую область памяти
    mas[n] = actor; //вносим новый элемент на последнюю позицию
    delete[]actors; //удаляем старый массив
    actors = mas; //перенаправляем указатель на новый увеличенный массив
    ++n;
    return true;
}
bool TMovie::deleteActor(string firstName_, string secondName_)
{
    if (!isExistdActor(firstName_, secondName_)) return false;
    TActor * mas = new TActor[n - 1]; //выделяем новую область памяти
    int j = 0;
    for (int i = 0; i < n; ++i)
    {
        if (actors[i].firstName == firstName_ && actors[i].secondName ==
secondName_)
        {
            continue;
        }
        mas[j++] = actors[i]; //переносим массив в новую область памяти
    }
    delete[]actors; //удаляем старый массив
    actors = mas; //перенаправляем указатель на новый увеличенный массив
    --n;
}
TActor*TMovie::findOldestActor()const
{
    if (n <= 0) return NULL;
    TActor* oldestActor = actors;
    for (int i = 1; i < n; ++i)
    {
        if (actors[i].yearBirth < oldestActor->yearBirth) oldestActor = &actors[i];
    }
}

```

```

        return oldestActor;
    }
    int TMovie::calcAverageActorsTime()const
    {
        int result = 0;
        for (int i = 0; i < n; ++i)
            result += (actors[i].yearFirstMove - actors[i].yearBirth);
        result /= n;
        return result;
    }

    TMovie& TMovie::operator =(const TMovie& obj)
    {
        if (this == &obj) return (*this);
        this->~TMovie();
        nameMovie = obj.nameMovie;
        country = obj.country;
        genre = obj.genre;
        ageaudience = obj.ageaudience;
        lengthOfFilm = obj.lengthOfFilm;
        n = obj.n;
        actors = new TActor[n];
        for (int i = 0; i < n; ++i)
        {
            actors[i] = obj.actors[i];
        }
    }

    TMovie& TMovie::operator +=(const TActor& actor)
    {
        addActor(actor);
        return(*this);
    }

    bool TMovie::operator < (const TMovie& arg2) const
    {
        return (this->lengthOfFilm < arg2.lengthOfFilm);
    }

    TMovie::operator int()const
    {
        int result = calcAverageActorsTime();
        return result;
    }

    ostream& operator <<(ostream& out, const TMovie& movie)
    {
        out << movie.getNameMovie()
            << "\t" << movie.getCountry()
            << "\t" << "length of film: " << movie.getLengthOfFilm()
            << "\t the age audience: " << movie.getAgeaudience()
            << "\tgenre : " << movie.getNameGenre() << endl;
        movie.outputActors(out);
        return out;
    }

    istream& operator >>(istream& in, TMovie& movie)
    {
        string strTmp;
        cout << "Please input name movie: ";
        getline(in, strTmp);
        movie.setNameMovie(strTmp);
        cout << "Please input country: ";
        getline(in, strTmp);
        movie.setCountry(strTmp);
        cout << "Please input length of film: ";
        getline(in, strTmp);
        movie.setLengthOfFilm(atoi(strTmp.c_str()));
    }

```

```

    cout << "Please input the age audience: ";
    getline(in, strTmp);
    movie.setAgeaudience(atoi(strTmp.c_str()));
    cout << "Please choose the genre from the following:";
    for (int i = 0; i < NUMBER_GENRE; ++i)
        cout << i << "\t" << listNameGetre[i] << endl;
    getline(in, strTmp);
    int intGetre = atoi(strTmp.c_str());
    Tgenre g = (intGetre >= 0 && intGetre <= NUMBER_GENRE)? (Tgenre)intGetre: ACTION;
    movie.setNameGenre(g);
    string answer;
    do
    {
        cout << "Please add actor:";
        TActor actor;
        in >> actor;
        movie += actor;
        cout << "Do wount continue add actors(y/n):";
        getline(in, answer);
    } while (answer.size() != 0 && answer[0] == 'y');
    return in;
}

```

```
c:\users\win7x64\source\repos\Film\Debug\Film.exe
Current Datetime:Mon Mar 19 21:35:15 2018
Please input name movie: Хоббит
Please input country: Украина
Please input length of film: 190
Please input the age audience: 18
Please choose the genre from the following:0 ACTION
1 WESTERN
2 MYSTERY
3 DRAMA
4 COMEDY
5 ROMANCE
6 ADVENTURE
7 THRILLER
8 HISTORY
9 HORROR
10 FANTASY
6
Please add actor:Please input first name: Иван
Please input second name: Иванов
Please input year of birth: 1978
Please input the year the first time starred in the film 2000
Do wount continue add actors(y/n):y
Please add actor:Please input first name: Никола
Please input second name: Петров
Please input year of birth: 1998
Please input the year the first time starred in the film 2003
Do wount continue add actors(y/n):n
Please add new actor:
Please input first name: Денис
Please input second name: Тесла
Please input year of birth: 2001
Please input the year the first time starred in the film 2010
Prepared info about film:
Хоббит Украина length of film: 0 the age audience: 18 genre :ADVENTURE

Иван Иванов 1978 2000
Никола Петров 1998 2003
Денис Тесла 2001 2010

Average Actors Time: 12
Average age actor: 25
Oldestactor is :
Иван Иванов 1978 2000
Please input name actor which need delete:
input first name: Денис
input second name: Тесла
Final info about film:
Хоббит Украина length of film: 0 the age audience: 18 genre :ADVENTURE

Иван Иванов 1978 2000
Никола Петров 1998 2003

Для продолжения нажмите любую клавишу . . . _
```

ПЕРЕЛІК ПОСИЛАНЬ

1. Методичні вказівки до лабораторних та практичних робіт з дисципліни “Об’єктно-орієнтоване програмування” за спеціальністю 122 «Комп’ютерні науки та інформаційні технології», освітня програма «Інформатика» для студентів освітньо-кваліфікаційного рівня «бакалавр» /Упоряд.: В.А. Любченко, О.В. Яковлева, Д.О. Руденко – Харків: ХНУРЕ, 2023. – 171с
2. Основи програмування мовою C++: Навчальний посібник /Є.П. Путятін, В.А. Любченко, О.А. Кобилін, Д.О. Руденко, Д.С. Пелешенко— Х.: ТОВ «Компанія СМІТ», 2018.— 260 с.
3. Грицюк Ю.І., Рак Т.Є.Г 85 Програмування мовою C++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с. – Статистика: іл. 10, табл. 18, бібліогр. 31(<http://xn--e1ajqk.kiev.ua/wp-content/uploads/2019/12/Griczyuk-C.pdf>)
4. C/C++ Programmer's Reference by Herbert Schildt (Fourth Edition)
5. C Programming Language (2nd Edition) By Brian W. Kernighan & Dennis M. Ritchie
6. C++ Primer (5th Edition) By Stanley B. Lippman, Josee Lajoie, and Barbara E Moo