

# Опис проекту на C++

Ільченко Єва

7 грудня 2023 р.

## 1 Опис коду

Ця програма на C++ моделює університетську спільноту, використовуючи класи Human-Людина, Worker-Працівник, Professor-Викладач, Student-Студент, Transcript-Заліковка і Profession-Професія.

Основна функція main використовує ці класи для створення об'єктів, додавання їх до вектору universityMembers, виведення інформації на екран, зберігання цієї інформації в файл university members.json, читання з файлу та роботу з динамічним поліморфізмом для обчислення середнього балу студентів, які вчить певний викладач.

### 1.1 Основні етапи програми:

1. Створення об'єктів працівника, професора і студента.
2. Додавання цих об'єктів до вектору universityMembers.
3. Виведення інформації про університетських членів у форматі JSON.
4. Зберігання цієї інформації в файл university members.json.
5. Читання та виведення збереженої інформації з файлу.
6. Обчислення середнього балу студентів певного викладача.
7. Звільнення пам'яті, видалення створених об'єктів.

### 1.2 Класи та функції:

1. Клас Human:
  - Описує базовий клас для усіх членів університетської спільноти.
  - Містить дві чисто віртуальні функції: display() (для виведення інформації) та дві функції для обчислення зарплати та стипендії.
2. Клас Worker:
  - Клас-нащадок Human, представляє працівників університету.
  - Містить поля для базової зарплати та досвіду роботи.
  - Реалізує віртуальні функції з класу Human.
3. Клас Professor:
  - Клас-нащадок Worker, надає інформацію про викладачів.
  - Додає поле для наукового ступеня.
  - Перевизначає функцію display(), розширюючи її виведення інформації.
4. Клас Student:
  - Клас-нащадок Human, надає інформацію про студентів.
  - Містить поле для середнього балу студента.

- Реалізує віртуальні функції з класу Human.

#### 5. Клас Transcript:

- Клас-нащадок Student. Надає інформацію про заліковку студента.
- Містить поля для імені студента та середнього балу.
- Має функцію `display()` для виведення інформації про заліковку.

#### 6. Клас Profession:

- Клас-нащадок Professor. Представляє професію для працівників.
- Містить поля для назви професії та базової зарплати.
- Має функцію `display()` для виведення інформації про професію.

#### 7. Функція `printJson`:

- Шаблонна функція для виведення вектора об'єктів у форматі JSON.
- Використовує функцію `display()` для кожного об'єкта.

#### 8. Функція `main`:

- Створює вектор вказівників на об'єкти класу Human.
- Додає до вектору об'єкти працівника, професора та студента.
- Виводить інформацію про університетських членів у форматі JSON за допомогою функції `printJson`.
- Зберігає інформацію в файл `university members.json`.
- Зчитує інформацію з файлу та виводить її на екран.
- Використовує динамічний поліморфізм для обчислення середнього балу студентів, яких вчить певний викладач.
- Звільняє пам'ять, видаляючи створені об'єкти.

## 2 Задачі на дослідження бібліотки

### 2.1 Обробка Винятків та Помилоч:

Можна додати обробку винятків для випадків, коли завантаження або збереження в файл не вдається. Використовуйте конструкції `try`, `catch` для обробки винятків та виведення повідомлень про помилку у відповідних випадках.

### 2.2 Розширення задачі для викладачів

Можна додати можливість обчислення середнього балу студентів для конкретного професора. Розширити клас `Professor` для зберігання списку студентів, які йому підпорядковані, і використати це для обчислення середнього балу.

### 2.3 Додавання нових об'єктів

Можна додати новий клас (наприклад, "Staff"), щоб розширити ієрархію класів. Додати об'єкти цього класу до списку `universityMembers` та вивести їх у форматі JSON.

### 2.4 Додаткові статистичні дані

Можна розширити програму для обчислення і виведення додаткових статистик, таких як середня зарплата в університеті, максимальний та мінімальний середній бал студентів і так далі.

## 2.5 Поліморфізм

Можна використовувати вказівники або посилання на об'єкти базового класу Human для роботи з об'єктами його похідних класів.

## 2.6 Розширення функціоналу

Можемо Розглянути можливості розширення функціоналу програми. Наприклад, додати можливість редагування даних про працівників, викладачів та студентів під час використання програми.

## 2.7 Дослідження бібліотеки json-c

- Можемо використати різні функції цієї бібліотеки та порівняти їхні можливості з альтернативними бібліотеками.
- Провести дослідження щодо різних методів *парсингу* (аналіз структури даних з метою вилучення необхідної інформації) та *серіалізації* (процес перетворення будь-якої структури даних у послідовність бітів) JSON-даних в бібліотеці json-c. Можна зробити оцінку швидкості та ефективності різних підходів до парсингу та серіалізації в залежності від об'єму даних.
- Спробувати виконати операції додавання, видалення та зміни об'єктів у JSON-структурі.
- Спробувати здійснити перетворення між об'єктами C++ та JSON-структурами.
- Можемо порівняти бібліотеку json-c з іншими доступними бібліотеками для роботи з JSON в C++. Порівняти їх швидкість, зручність використання та інші критерії.

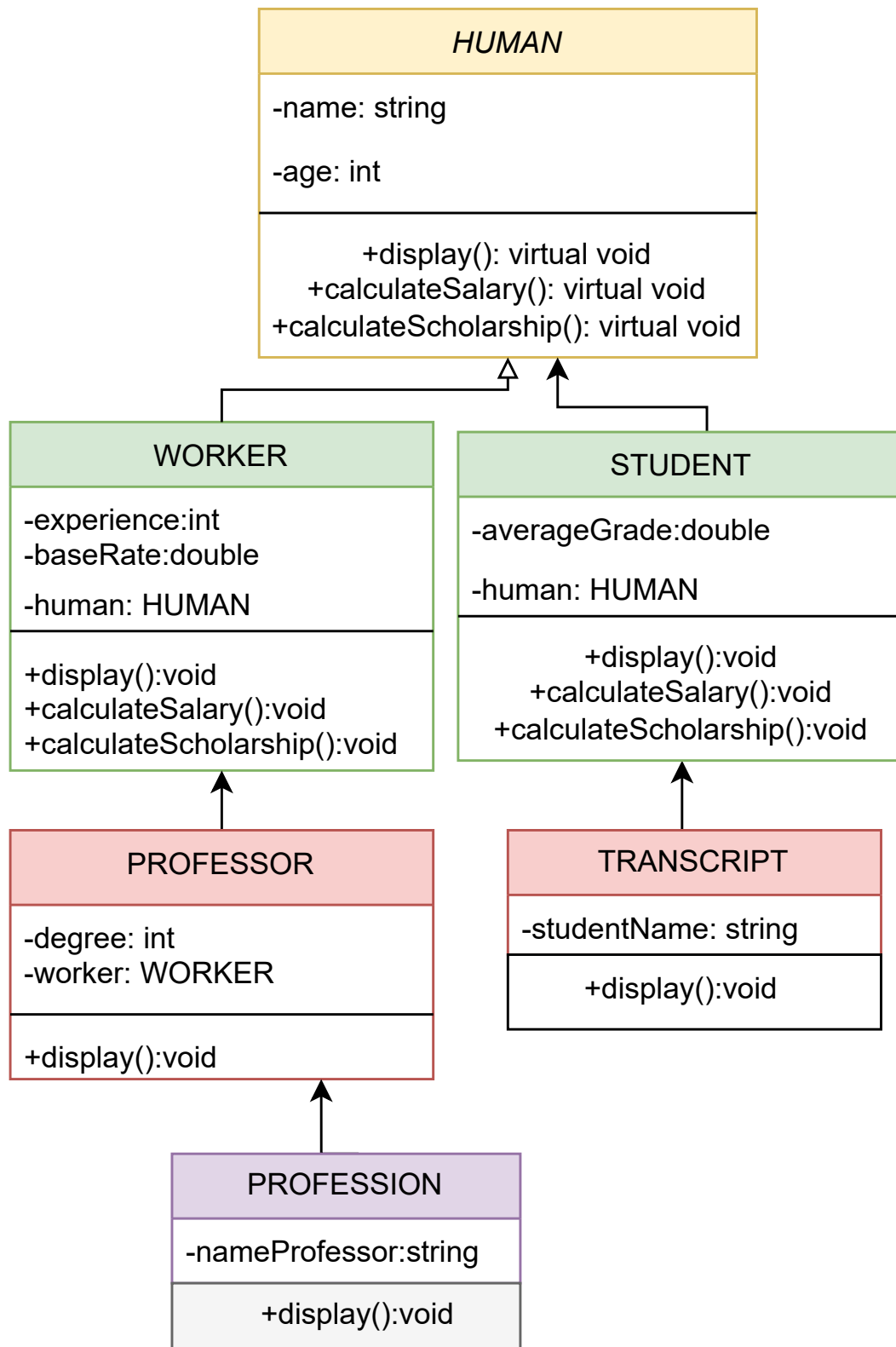


Рис. 1: UML-діаграма класів