**Class Room**

Private:

Size\_t numberOfSeats;

Std::string name;

**Class Movie**

Private:

std::string name;

std::string genre;

size\_t duration;

size\_t yearOfPremiere;

double price;

**Class Person**

Protected:

Cinema\* cinema;

Std::string names;

Std::String dateOfBirth;

Public:

Virtual void buyTicket(Screening\* screening) = 0;

Void getReservations();

Virtual ~Person() = default;

**Class Client : public Person**

Private:

Double balance;

enum discount;

public:

void buyTicket(Screening\* screening);

void addMoney(double money);

**Enum Discount**

{

None, elderly, school, student

}

**Class Employee : public Person**

Private:

Size\_t salary;

Public:

Void addMovie(const Movie& movie);

Void addScreening(const Screening& screening;

void buyTicket(Screening\* screening);

**Class Cinema**

Private:  
double balance;

Std::vector<Client> clients;

Std::vector<Employee> workers;

Size\_t workerId;

Timetable\* timetable;

Std::vector<Movie> films;

Public:

Void addMovie(size\_t workerId, Movie film);

Void addScreening(size\_t workerId, Screening screening);

Void addClient(Client client);

Void addWorker(Employee worker);

Void changeBalance(double money);

**Class Screening**

Private:

Movie\* film;

Room\* room;

Size\_t takenSeats;

Vector<Person\*> people;

Double price;

Public:

Void makeReservation(const Person\*);

Void print() const;

Double getPrice() const;

**Class Timetable**

Private:

Std::vector<Screening> screenings;

Public:

Void addScreening(Screening screening);