```
// !Ticket.h

#pragma once
#include "../Entities/Session.h"
#include "../Utils/Place.h"
class Ticket { private:
Session *session;
Place *place;
public:
Ticket(Session* session, Place* place);
Place* getPlace();
Session* getSession();
void print();
};

// !Ticket.cpp

#include "../Entities/Ticket.h"
Ticket::Ticket(Session* session, Place* place) {
this->session = session;
this->place = place;
}
Place* Ticket::getPlace() {
return this->place;
}
Session* Ticket::getSession() {
return this->session;
}
void Ticket::print() {
this->session->print();
this->place->print();
}

// !Session.h

#pragma once
#include "../Utils/Time.h"
#include "iostream"

class Session {
private:
Time startTime;
Time endTime;

std::string name;
int hole;
public:
explicit Session();

Session(Time startTime, Time endTime, std::string name);

Time getStartTime();

Time getEndTime();

std::string getName();

int getHole();

void print();
};
```