

Задача5_3 Макрос UPDATE_FIELD

Продолжим работу над сервисом поиска авиабилетов AviaScanner. Наш сервис хранит базу данных билетов в виде `vector<AirlineTicket>`, где `AirlineTicket` — такая же структура, как и в предыдущей задаче. Периодически наш сервис обходит сайты авиакомпаний, собирает свежую информацию о доступных билетах и обновляет записи в своей базе данных. Делается это с помощью функции `void UpdateTicket(AirlineTicket& ticket, const map<string, string>& updates)`. Параметр `updates` содержит пары (имя поля; новое значение). При этом он содержит только те поля, которые поменялись. Пример работы функции `UpdateTicket`:

```
1 void UpdateTicketExample() {
2   AirlineTicket t;
3   t.price = 5000;
4   t.from = "DME";
5   t.to = "AER";
6
7   const map<string, string> updates = {
8     {"price", "3500"},
9     {"from", "VKO"}
10  };
11  UpdateTicket(t, updates);
12  ASSERT_EQUAL(t.from, "VKO");
13  ASSERT_EQUAL(t.to, "AER");
14  ASSERT_EQUAL(t.price, 3500);
15 }
```

Функцию `UpdateTicket` можно было бы реализовать так:

```
1 void UpdateTicket(AirlineTicket& ticket, const map<string, string>& updates) {
2   map<string, string>::const_iterator it;
3
4   it = updates.find("to");
5   if (it != updates.end()) {
6     ticket.to = it->second;
7   }
8
9   it = updates.find("from");
10  if (it != updates.end()) {
11    ticket.from = it->second;
12  }
13
14  it = updates.find("price");
15  if (it != updates.end()) {
16    istringstream is(it->second);
17    is >> ticket.price;
18  }
19
20  ...
21 }
```

Здесь налицо дублирование кода — блоки кода внутри функции `UpdateTicket` отличаются только именем поля (конечно, первые два блока не такие, как третий, но их легко к нему свести). При этом имя поля используется не только для обращения к структуре `AirlineTicket`, но и как строковый литерал. Поэтому можно написать макрос, который существенно упростит функцию `UpdateTicket`:

```
1  #define UPDATE_FIELD(ticket, field, values) ...
2
3  void UpdateTicket(AirlineTicket& ticket, const map<string, string>& updates) {
4  UPDATE_FIELD(ticket, to, updates);
5  UPDATE_FIELD(ticket, from, updates);
6  UPDATE_FIELD(ticket, price, updates);
7  UPDATE_FIELD(ticket, airline, updates);
8  ...
9  }
```

Напишите макрос `UPDATE_FIELD`. Вам дан файл `airline_ticket.h`, содержащий объявления структур `Time`, `Date` и `AirlineTicket`, а также заготовка решения в виде `cpp`-файла `update_field.cpp`. Пришлите на проверку `cpp`-файл, который

- подключает заголовочный файл `airline_ticket.h`
- содержит макрос `UPDATE_FIELD`
- содержит определения операторов, необходимых для считывания классов `Date` и `Time` из потока `istream` и их использования в макросе `ASSERT_EQUAL` (формат ввода смотрите в юнит-тестах в файле `update_field.cpp`)

update_field.cpp

airline_ticket.h