

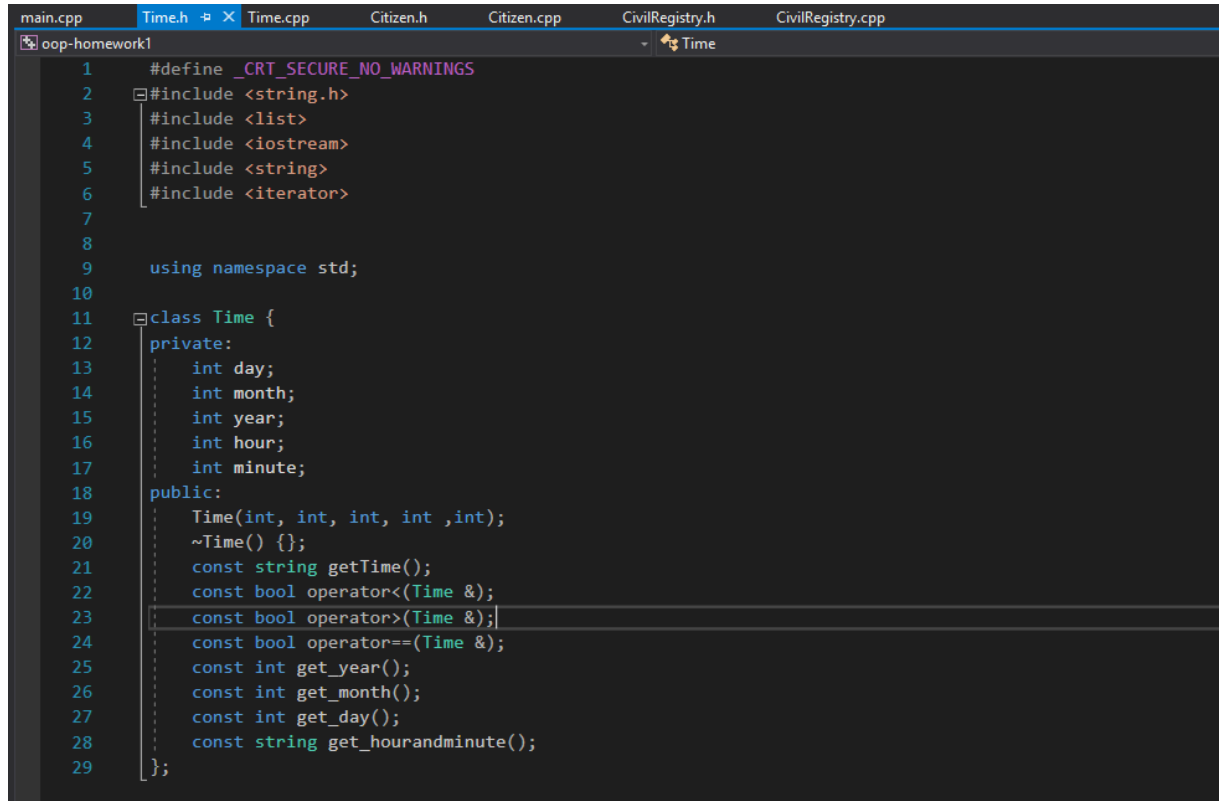
OOP Homework 1 Report

Mutlaka dikkat edilmesi gereken nokta:

Derlerken şu şekilde derlenmesi gerekiyor:

```
g++ -std=c++11 *.cpp -o output
```

stoi kullanıldığı için derlerken yukarıdaki gibi derleyin lütfen...



```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <string.h>
3  #include <list>
4  #include <iostream>
5  #include <string>
6  #include <iterator>
7
8
9  using namespace std;
10
11 class Time {
12 private:
13     int day;
14     int month;
15     int year;
16     int hour;
17     int minute;
18 public:
19     Time(int, int, int, int ,int);
20     ~Time() {};
21     const string getTime();
22     const bool operator<(Time &);
23     const bool operator>(Time &);
24     const bool operator==(Time &);
25     const int get_year();
26     const int get_month();
27     const int get_day();
28     const string get_hourandminute();
29 };
```

Şekil 1 Time.h

Fonksiyonların dönüş değerleri const olması gerektiği için Şekil1'deki gibi tanımlanmıştır ama operatör fonksiyonlarının argümanları atama işleminde kullanıldığı için const yapılmamıştır. Destructor kısmında alan alınmadığı için herhangi bir silme işlemine gerek yoktur.

```
main.cpp  Time.h  Time.cpp  Citizen.h  Citizen.cpp  CivilRegistry.h  CivilRegistry.cpp
oop-homework1  Citizen
1  #include "Time.h"
2
3  using namespace std;
4
5  class Citizen {
6  private:
7      char *idNo;
8      char *name;
9      char *surname;
10     bool hasApp;
11     Time *appTime;
12     char *appSlot;
13 public:
14     Citizen(char *, char *, char *, char *, char *, char *,char *);
15     ~Citizen();
16     Time& getAppTime();
17     const bool gethasApp();
18     const string getAppSlot();
19     const string get_name();
20 };
21
```

Şekil 2 Citizen.h

GetAppTime fonksiyonu Time objesi döndürmektedir ve bu obje atamalarda kullanıldığı için const yapılmamıştır geri kalan fonksiyonlarda buna benzer bir kısıtlama olmadığı için hepsi const yapılmıştır. Destructor kısmında dinamik olarak alınan bütün alanlar geri verilmiştir.

```
main.cpp  Time.h  Time.cpp  Citizen.h  Citizen.cpp  CivilRegistry.h  CivilRegistry.cpp
oop-homework1  CivilRegistry
1  #include "Citizen.h"
2
3  using namespace std;
4
5  class CivilRegistry {
6  private:
7      list <Citizen> wApp;
8      list <Citizen> wOutApp;
9
10 public:
11     CivilRegistry() {}
12     ~CivilRegistry();
13     void insertCitizen(Citizen &);
14     void removeCitizen(int);
15 };
16
```

Şekil 3 CivilRegistry.h

Constructor kısmı herhangi bir atam gerektirmediği için boş bırakılmıştır ama destructor kısmında list değişkenlerinin içleri boşaltılmıştır. insertCitizen fonksiyonu const değişken alamadığı

için Şekil 3'deki gibi tanımlanmıştır.removeCitizen fonksiyonun içinde listelerin içindekiler yazılmış ve listeler poplanmıştır.

Genel tanımlamalar:

İnputlar okunurken dinamik olarak okunmuştur ve okunan inputların max boyutları main.cpp dosyasının en başında define olarak tanımlanmıştır. Her satırda dinamik olarak Citizen objesi oluşturulmuş ve bu insert fonksiyonuna gönderilerek listeye eklenmiştir.

Class tanımlamaları:

Bütün classlardaki datalar private yapılmıştır bu datalara get methodlarıyla ulaşılmıştır.Böylece kod daha güvenli hale gelmiştir. CivilRegistry clasında da time değerlerine ulaşmak için getAppTime fonksiyonu bize Time objesi dönmektedir böylece classlarda friend yapmadan veri aktarımı sağlanmıştır. Atama olmayan bütün değişkenler ve methodlar const yapılmıştır böylece daha güvenli bir program yazılmıştır.