



INF 212 Algorithms and Programming II

**2020-2021 Spring
Electronic Engineering**

Project 4

DETECTİVE GAME

Team Members

School ID	Name	Surname
171024009	ÖMER	GENÇ
171024091	UĞUR	KILINÇ
171024095	MUHAMMET ALİ	KILIÇ
1801022104	ERKAN	BIÇAKÇI

PROJECT OBJECTIVE

Projemiz “Detective Game” olarak adlandırılmaktadır. Mevcut olay örgüsüne uygun olarak düzenlenmiş hikâyeye uygun sorular sorularak verilen cevaplar üzerinden olay örgüsünün tamamlanması hedeflenmektedir. İsterseniz kendi hikayenizi, olay örgünüzü ve sorularınızı hazırlayabilme imkânınız da mevcuttur. Mantıklı ve mantıksız sorular sorarak amacımız oyunu tamamlamaktır.

PROJECT

Öncelikle yazar modu kısmında hikaye, olay örgüsü ve sorular eklenebilir. Dedektif oyununa uygun bir şekilde gizemli hikayeler oluşturulabilir. Bu hikayelere göre sorular yazılabilir.

Kayıtlı oyunlar seçeneğinde ise daha önceden koda uyarlanmış farklı hikayelerin bulunduğu başka dedektif oyunlarının eklendiği kısımdır. Fakat biz sadece 1 tane oyun yükledik istenilirse başka oyunlar yüklenilerek kayıtlı oyunlar bölümünde istediğiniz oyun oynanabilir.

Yüklü oyun ise bizim daha önce programa eklediğimiz oyun mevcuttur. Bu kısma girilerek kayıt edilmiş oyun oynanmaktadır.

Çıkış bölümünde ise isteğe bağlı olarak programdan çıkış yapılmaktadır.

DESIGN ALGORITHM**1) Oynanış**

Oynanış ana class olan Case classı üzerine kurulmuştur. Case classı içerisinde 1 adet Detective 1 adet de Person nesnesi bulunmaktadır. Person classı içerisinde ise 5 adet Question nesnesi bulunmaktadır. Case metodu üzerinden Play metodu çalıştırıldığında ilgili nesneye ait senaryo oynanmaktadır. İlk başta Case nesnesi içerisinde ki story kısmı ekrana yazdırılır ve oyun başlar. Daha sonra Person nesnesi içerisinde ki Question nesnesinin birincisi çağrılır ve ekrana 4 soru basılır. Bu 4 sorudan 1'i gereksiz 3 ü ise hikayeye katkısı olan sorulardır. Gerekli olan 3 soru cevaplandığında Question nesnesinin de ki story ekrana basılır. Bu kısım dedektifimizi birazcık yönlendirir ve sonraki soru seti çağrılır. Bu şekilde 5 set çağrılır. Amaç gereksiz sorulara takılmadan finale ulaşmaktır. Finalde ise Case nesnesi içerisinde ki final ekrana bastırılır ve sır perdesi aralanmış olunur.

2) Dosyaya yazma

Dosyaya yazma işlemi için Case nesnesi içerisinde de ki Case_WriteToFile metodu ile gerçekleşir. Case verileri diyagramda gösterildiği gibi parçalanarak farklı dosyalara yazılır. Farklı dosyalara yazılmasının da ki amaç ise okuma işlemini kolaylaştırmaktır.

3) Dosyadan okuma

Dosyadan okuma işlemleri için gerekli olan fonksiyonlar File .h ve File.cpp dosyası içerisine yazıldı. Burada birçok farklı işlevi olan fonksiyon var kimisi soru setlerini okumaktan görevli kimileri ise Person verilerini okumakla görevli. Bu fonksiyonlar bir araya geldiklerinde, istediğimiz senaryoya uygun olan verileri çekiyor ve bir Case nesnesine aktarıyor. Daha sonra Play metodunu çağırarak oyunu oynayabilmemizi sağlıyor.

3)Yazar modu

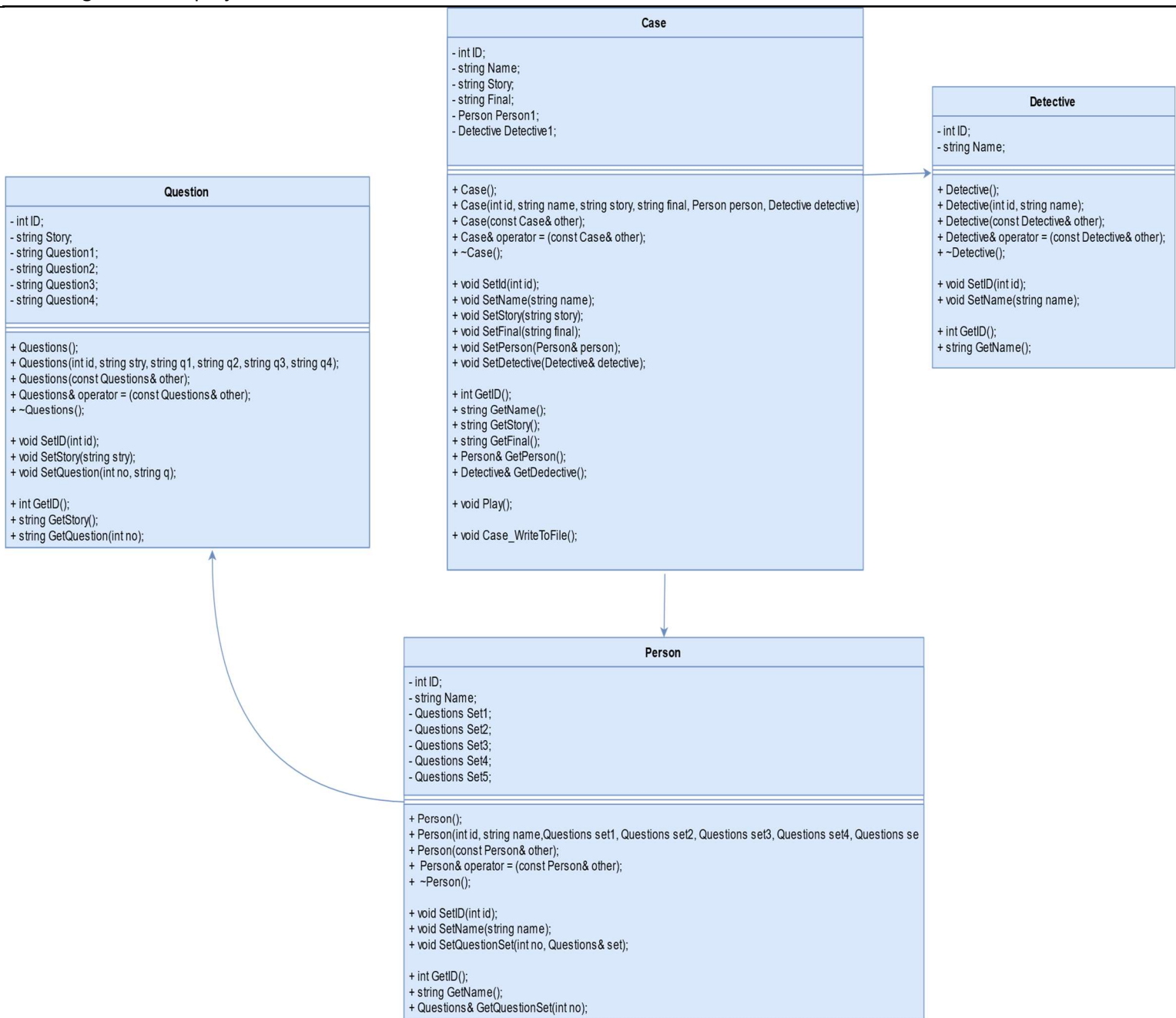
Bu mod sayesinde yazarlar kendi senaryolarını oyuna dönüştürme şansına sahip oluyorlar. Bir Case nesnesi oluşturuluyor ve veriler kullanıcıdan alınıyor. Daha sonra dosyaya yazma metodu ile kayıt altına alınıyor.

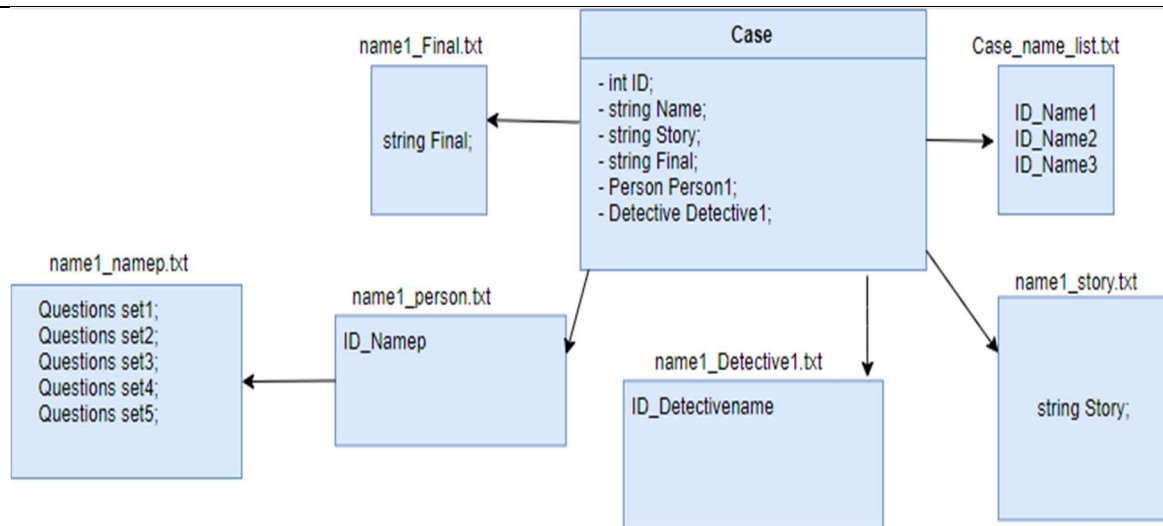
4) Çıkış

Programı istenilen zamanda sonlandırılmaktadır.

UML DIAGRAM

UML diagram of the project





HEADER FILE(s)

Hedheader file(s) of the project

```

#ifndef CASE_H_
#define CASE_H_
#include "Questions.h"
#include "Person.h"
#include "Detective.h"
#include <fstream>
#include <string>
#include <cstring>

class Case
{
friend ostream& operator << (ostream& os, const Case _Case);
private:
    int ID;
    string Name;
    string Story;
    string Final;
    Person Person1;
    Detective Detective1;
public:
    Case();
    Case(int id, string name, string story, string final, Person person, Detective detective);
    Case(const Case& other);
    Case& operator = (const Case& other);
    ~Case();

    void SetId(int id);
    void SetName(string name);
    void SetStory(string story);
    void SetFinal(string final);
    void SetPerson(Person& person);
    void SetDetective(Detective& detective);
  
```

```
int GetID();
string GetName();
string GetStory();
string GetFinal();
Person& GetPerson();
Detective& GetDetective();

void Play();
```

```
//Dosyaya yazma işlemleri Uğur kılınç
void Case_WriteToFile();
```

```
};
ostream& operator << (ostream& os, const Case _Case);
Case operator + (Case _Case1, Case _Case2);
```

```
void Devam();
void Cevap1(int c1, int c2, int c3, int c4);
void Cevap2(int c1, int c2, int c3, int c4);
void Cevap3(int c1, int c2, int c3, int c4);
void Cevap4(int c1, int c2, int c3, int c4);
void Cevap5(int c1, int c2, int c3, int c4);
```

```
void caseDuzenle(Case& other);
```

```
#endif
```

```
#ifndef DETECTIVE_H_
#define DETECTIVE_H_
#include<iostream>
using namespace std;
```

```
class Detective
{
    friend ostream& operator << (ostream& os, const Detective detec);
private:
    int ID;
    string Name;
public:
    Detective();
    Detective(int id, string name);
    Detective(const Detective& other);
    Detective& operator = (const Detective& other);
    ~Detective();

    void SetID(int id);
```

```
void SetName(string name);

int GetID();
string GetName();

};
ostream& operator << (ostream& os, const Detective detec);
Detective operator + (Detective detec1, Detective detec2);
```

```
#endif
```

```
#ifndef Exception_h
#define Exception_h
```

```
#include <iostream>
#include <string>
using namespace std;
```

```
class Exception{
public:
Exception(string err_message){
    message = err_message;
}
```

```
string getMessage(){
    message = "Hata! Girilen deger "+ message;
    return message;
}
private:
string message;
};
```

```
#endif
```

```
#ifndef FILE_H_
#define FILE_H_
#include "Case.h"
#include<iostream>
#include<fstream>
#include<string>
#include<cstring>
using namespace std;
```

```
string Case_Read_ID_Name();
int getCaseID();
string getCaseName();

string Case_Read_Story(string name);
string Case_Read_Final(string name);

string Case_Read_Detective(string name);
int getDetectiveID(string metin);
string getDetectiveName(string metin);

string Case_Read_Person(string name);
int getPersonID(string metin);
string getPersonName(string metin);

string Case_Read_Set_Story(string name, int set_ID);
string Case_Read_Quesitons(const string name, int q_ID);

void readCase(Case &senaryo);
#endif

#ifndef PERSON_H_
#define PERSON_H_
#include "Questions.h"

class Person{
    friend ostream& operator << (ostream& os, const Person person);
private:
    int ID;
    string Name;
    Questions Set1;
    Questions Set2;
    Questions Set3;
    Questions Set4;
    Questions Set5;
public:
    Person();
    Person(int id, string name, Questions set1, Questions set2, Questions set3, Questions set4, Questions set5);
    Person(const Person& other);
    Person& operator = (const Person& other);
    ~Person();

    void SetID(int id);
    void SetName(string name);
    void SetQuestionSet(int no, Questions& set); // soru seti no girilerek yapılacak karar yapıları kullan

    int GetID();
    string GetName();
    Questions& GetQuestionSet(int no); //soru seti no girilerek yapılacak karar yapıları kullan
```



```
};
ostream& operator << (ostream& os, const Person person);
Person operator + (Person person1, Person person2);

#endif

#ifndef QUESTIONS_H_
#define QUESTIONS_H_
#include <iostream>
using namespace std;

class Questions
{
    friend ostream& operator << (ostream& os, const Questions que);
private:
    int ID;
    string Story;
    string Question1;
    string Question2;
    string Question3;
    string Question4;
public:
    Questions();
    Questions(int id, string stry, string q1, string q2, string q3, string q4);
    Questions(const Questions& other);
    Questions& operator = (const Questions& other);
    ~Questions();

    void SetID(int id);
    void SetStory(string stry);
    void SetQuestion(int no, string q); // soru numarası ve ssoru girerek yapılacak else veya switc case kullan

    int GetID();
    string GetStory();
    string GetQuestion(int no); // soru numarası girilerek soru alacak esl if yapısı veya switc case kullan
};

ostream& operator << (ostream& os, const Questions que);
Questions operator + (Questions q1, Questions q2);

#endif

#ifndef TEST_H_
#define TEST_H_
#include "Case.h"

void TestDedective();
```

```
void TestQuestions();  
void TestPerson();  
#endif
```

TESTING

Briefly describe how you test your code

```
1. Yazar modu  
2. Kayitli oyunlar  
3. Yüklü oyun  
4. Cikis  
Secim: █
```

```
1. Yazar modu  
2. Kayitli oyunlar  
3. Yüklü oyun  
4. Cikis  
Secim: 1  
Senaryo ismi gir: Mali  
Senaryo ID gir: 2  
Senaryo hikayesi gir: Vadide_Dehşet  
Senaryo finali gir: Suçlu_Bulundu  
Dedektif ismi gir: Can  
Dedektif ID gir: 3  
Soru seti 1 için story gir1  
Soru seti 2 için story gir2  
Soru seti 3 için story gir3  
Soru seti 4 için story gir4  
Soru seti 5 için story gir5  
Soru seti 1  
Soru 1 giriniz: Suçluyu_gördün_mü █
```

```
Soru 2 giriniz: Soru 3 giriniz: 2
Soru 4 giriniz: 3
Soru seti 2
Soru 1 giriniz: 4
Soru 2 giriniz: 5
Soru 3 giriniz: 6
Soru 4 giriniz: 7
Soru seti 3
Soru 1 giriniz: 2
Soru 2 giriniz: 3
Soru 3 giriniz: 4
Soru 4 giriniz: 5
Soru seti 4
Soru 1 giriniz: 1
Soru 2 giriniz: 2
Soru 3 giriniz: 3
Soru 4 giriniz: 4
Soru seti 5
Soru 1 giriniz: 1
Soru 2 giriniz: 2
Soru 3 giriniz: 3
Soru 4 giriniz: 4
Person ID gir: 2
Person Name gir: Erkan
```

1. Yazar modu
2. Kayıtlı oyunlar
3. Yüklü oyun
4. Çıkış

Secim: 3

Orkun hafta sonunu değerlendirmek için sanat galerisine gider. Sanat galerisinde gezinirken bir süre sonra sıkılı r ve çıkmaya karar verir ve çıkmadan önce son kez o fotoğrafa bakmak ister. Çok ilginç bir fotoğraf, tamamen siya h beyaz eski bir fotoğraf. Arka planı çok belli değildir ve mekânı kestirmek neredeyse imkansızdır. Fotoğrafı ilg inç kılan şey ise karşılıklı oturan iki ceset yetmezmiş gibi bir de masanın üzerinde bir satranç tahtası olmasıdı r. Bu fotoğrafın sırrını öğrenmek isteyen Orkun galeri görevlisi ile konuşur ve fotoğrafın izini sürmeye karar v erir. Edindiği bilgiye göre bu fotoğraf galeriye emekli bir donanma amirali tarafından başışlanmıştır. Uzun süren araştırmalar sonucunda bu amiral ile görüşme ayarlar. Görüşmeye giden Orkun hiç ummadığı bir şey ile karşılaşır. Amiral çok yaşlıdır ve konuşma yazma gibi motor fonksiyonlarını kaybetmiştir. Mektuplaştığı kişi yani amiralin e şi ona şunları söyler. "Biliyorum umduğunu bulamadın bende o fotoğrafın sırrını bilmiyorum, ne kadar konuşamasa d a bazen soru sorduğumda gözleri ile evet ve hayır cevapları verebilir. Sağ gözünü kırparsa evet sol gözünü kırpar sa hayır demek istiyordur." bunu üzerine Orkun evet ve hayırlı sorular sormaya başlar.... devam etmek için herhangi bir deger giriniz....

1. Üzerinde üniforma bulunan bu kişiler asker mi?
2. Bu kişiler sizin askerleriniz miydi?
3. Olay bir gemide mi yaşandı?
4. Olay denizde mi yaşandı ?
Lutfen soru seciniz: 1
Evet
Lutfen soru seciniz: 2
Evet
Lutfen soru seciniz: 3
Hayır
Lutfen soru seciniz: 4
Evet
Lutfen soru seciniz: 5
Hatalı secim..
Lutfen soru seciniz: 3
Hayır

Olay bir deniz taşıtında yaşanmış olmalı sonuçta masa ve satranç tahtası var. Acaba ne tür bir taşıt tekne olması çok olası bir durum...

devam etmek için herhangi bir deger giriniz....1

1. Olay vir deniz taşıtında mı yaşandı?
2. Olay bir teknede mi yaşandı?
3. Askerler birileri tarafından mı öldürüldü?
4. Bu kişiler dost muydu?

Lutfen soru seciniz: 1

Evet

Lutfen soru seciniz: 2

Hayır

Lutfen soru seciniz: 3

Hayır

Lutfen soru seciniz: 4

Evet

Bu kişiler bir başkası tarafından öldürülmemiş, amiralin ölümler ile bir alakası olduğunu düşünmüştüm. Galiba yan ıldım, tabi yalan söylemiyorsa...

devam etmek için herhangi bir deger giriniz....2

1. Askerler birbirlerini mi vurdu?
2. Fotoğrafta gözükmeyen 3. bir şahıs var mıydı?
3. Ölmeden önce satranç mı oynadılar?
4. Ölümünün oyun ile alakası var mı?

Lutfen soru seciniz: 1

Hayır

Lutfen soru seciniz: 2

Hayır

Lutfen soru seciniz: 3

Evet

Lutfen soru seciniz: 4

Evet

Yaşlı amiral gözlerini odasında asılı duran silaha yöneltir. Bunlar silah ile mi öldü diye sorduğunda iki közü nü de kapatır. Bu hem evet hem de hayır mı demek istedi, peki ama bunun anlamı ne?

devam etmek için herhangi bir deger giriniz....3

1. Askerlerden biri silah ile vurularak mı öldü ?
2. Silahla ölen kişi kaza sonucu mu öldü?
3. Silah ile ölen kişi intihar mı etti?
4. Deniz taşıtında olduğunu söylemiştin bu taşıt deniz altı mı?

Lutfen soru seciniz: 1

Evet

Lutfen soru seciniz: 2

Hayır

Lutfen soru seciniz: 3

Evet

Lutfen soru seciniz: 4

Evet

Askerler, satranç, intihar... taşlar şimdi yerine oturdu
devam etmek için herhangi bir değer giriniz....5

Her şey anlaşıldı bu iki asker deniz altında görevli imiş, taşıtları hasar görmüş ve batıyormuş. Askerler öleceklerini anlayınca bir satranç oyunu oynamaya karar verirler. Kazanan kişi daha acısız ve kolay olan yolu yani silah ile intihar etmeye hak kazanmış. Kazanan kişi intihar eder kaybeden ise boğularak ölür. İyi ama diğeri neden silahlı kendini vurmadı? Çünkü tek bir mermileri vardı. Bu sözler bizzat amiralin eşine aittir. Orkun döner ve amiralin eşine konuşur, ama bilmediğini söylemişsiniz der. Bu olayı çözen 5. kişinin der amiralin eşi. İnsan yaşlanınca geleni gideni azalıyor. Ben ve eşim böyle bir oyun oynamaya karar verdik. Böylece son günlerimizde ziyaretçilerimiz çok oluyor bu sayede kendimizi yalnız hissetmiyoruz. Eşim yıllar önce çekti bu fotoğrafı. Olayın detaylarını ise boğulan askerin bıraktığı bir nottan öğrendik.

USER'S GUIDE

Menümüz 4 bölümden oluşmaktadır.

- 1) Yazar Modu: Bu seçim kendi hikayenizi , sorularınızı ,kişilerin oluşturulduğu bölümdür.
- 2) Kayıtlı Oyunlar: Bu seçim kaydedilen oyunları okuma işlevini görür.
- 3) Yüklü Oyun: Bu seçim sisteme kaydedilmiş oyunu oynamamız için gerekli ekranın açılmasını sağlar.
- 4) Çıkış: Bu seçim programı sonlandırır.

CONCLUSION AND REMARKS

Programımız istenilen şekilde çalışmaktadır. Programımız kullanıcı dostudur. Programı geliştirmek için çeşitli fonksiyonlar eklenebilir.

Programda çeşitli sayıda olay örgüsü eklenip oynanabilir şekilde kurgulanmıştır.

Kullanıcıdan veri girişi olarak dosyaları dinamik hale getirdik.

Fonksiyonları kullanarak istenilen işlemleri daha sade ve basit hale getirdik.

REFERENCES

Put the list of references and sources (books, links to websites, videos, etc), which you used for project.

<https://stackoverflow.com/questions/14265581/parse-split-a-string-in-c-using-string-delimiter-standard-c>

https://www.tutorialspoint.com/cplusplus/cpp_classes_objects.htm

<https://slideplayer.biz.tr/slide/3627075/>

<https://turkmuhendis.net/cppprogramlama/dosya-islemleri/>

APPENDIX

PROJECT CODES

Put here code of your project.

```
#include "Case.h"
```

```
Case::Case(){
    Person prs;
    Detective dtc;

    ID = 0;
    Name = "Case";
    Story = "Story";
    Final = "Final";
    Person1 = prs;
    Detective1 = dtc;
}
```

```
Case::Case(int id, string name, string story, string final, Person person, Detective
detective){
    ID = id;
    Name = name;
    Story = story;
    Final = final;
    Person1 = person;
    Detective1 = detective;
}
```

```
Case::Case(const Case& other){
    ID = other.ID;
    Name = other.Name;
    Story = other.Story;
    Final = other.Final;
    Person1 = other.Person1;
    Detective1 = other.Detective1;
}
```

```
Case& Case::operator = (const Case& other){
    ID = other.ID;
    Name = other.Name;
    Story = other.Story;
    Final = other.Final;
    Person1 = other.Person1;
    Detective1 = other.Detective1;
    return *this;
}
```

```
Case::~~Case(){} 
```

```
void Case::SetId(int id){ID = id;}
```



```
void Case::SetName(string name){Name = name;}

void Case::SetStory(string story){Story = story;}

void Case::SetFinal(string final){Final = final;}

void Case::SetPerson(Person& person){Person1 = person;}

void Case::SetDetective(Detective& detective){Detective1 = detective;}

int Case::GetID(){return ID;}

string Case::GetName(){return Name;}

string Case::GetStory(){return Story;}

string Case::GetFinal(){return Final;}

Person& Case::GetPerson(){return Person1;}

Detective& Case::GetDedective(){return Detective1;}

ostream& operator << (ostream& os, const Case _Case){
    os<<"Name: "<< _Case.Name<<
    endl<<"ID: "<< _Case.ID<<
    endl<<"Detective: "<<endl<<_Case.Detective1<<
    endl<<"Story: "<< _Case.Story<<
    endl<<"Final: "<< _Case.Final<<
    endl<< _Case.Detective1<<
    endl<< _Case.Person1<<endl;

    return os;
}

Case operator + (Case _Case1, Case _Case2){
    int id = _Case1.GetID() + _Case2.GetID();
    string name = _Case1.GetName() + _Case2.GetName();
    string story = _Case2.GetStory() + _Case2.GetStory();
    string final = _Case1.GetFinal() + _Case2.GetFinal();
    Person person;
    Detective detective;
    Case x(id,name,story,final,person,detective);
    return x;
}

void Case::Play(){
    cout<<Story<<endl;
```



```
Devam();
int puan = 0;

// Soru soruluyor
string secim;
cout<<"1. "<<Person1.GetQuestionSet(1).GetQuestion(1)<<
endl<<"2. "<<Person1.GetQuestionSet(1).GetQuestion(2)<<
endl<<"3. "<<Person1.GetQuestionSet(1).GetQuestion(3)<<
endl<<"4. "<<Person1.GetQuestionSet(1).GetQuestion(4)<<endl;
Cevap1(1,1,0,1);
cout<<Person1.GetQuestionSet(1).GetStory()<<endl;
Devam();

// set2 sorulari
cout<<"1. "<<Person1.GetQuestionSet(2).GetQuestion(1)<<
endl<<"2. "<<Person1.GetQuestionSet(2).GetQuestion(2)<<
endl<<"3. "<<Person1.GetQuestionSet(2).GetQuestion(3)<<
endl<<"4. "<<Person1.GetQuestionSet(2).GetQuestion(4)<<endl;
Cevap2(1,0,0,1);
cout<<Person1.GetQuestionSet(2).GetStory()<<endl;
Devam();

//set3 sorulari
cout<<"1. "<<Person1.GetQuestionSet(3).GetQuestion(1)<<
endl<<"2. "<<Person1.GetQuestionSet(3).GetQuestion(2)<<
endl<<"3. "<<Person1.GetQuestionSet(3).GetQuestion(3)<<
endl<<"4. "<<Person1.GetQuestionSet(3).GetQuestion(4)<<endl;
Cevap3(0,0,1,1);
cout<<Person1.GetQuestionSet(3).GetStory()<<endl;
Devam();

//set4 sorulari
cout<<"1. "<<Person1.GetQuestionSet(4).GetQuestion(1)<<
endl<<"2. "<<Person1.GetQuestionSet(4).GetQuestion(2)<<
endl<<"3. "<<Person1.GetQuestionSet(4).GetQuestion(3)<<
endl<<"4. "<<Person1.GetQuestionSet(4).GetQuestion(4)<<endl;
Cevap4(1,0,1,1);
cout<<Person1.GetQuestionSet(4).GetStory()<<endl;
Devam();

//set5 sorulari
cout<<"1. "<<Person1.GetQuestionSet(5).GetQuestion(1)<<
endl<<"2. "<<Person1.GetQuestionSet(5).GetQuestion(2)<<
endl<<"3. "<<Person1.GetQuestionSet(5).GetQuestion(3)<<
endl<<"4. "<<Person1.GetQuestionSet(5).GetQuestion(4)<<endl;
Cevap5(1,1,1,1);
cout<<Person1.GetQuestionSet(5).GetStory()<<endl;
Devam();

cout<<Final;
```

```
}

void Devam(){
    string devam;
    cout<<"devam etmek icin herhangi bir deger giriniz....";
    cin>>devam;
}

void Cevap1(int c1, int c2, int c3, int c4)
{
    string C1 = "Evet",C2 = "Evet",C3 = "Evet",C4 = "Evet";
    int puan = 0;
    string secim;

    if(c1 == 0)
        C1 = "Hayır";
    if(c2 == 0)
        C2 = "Hayır";
    if(c3 == 0)
        C3 = "Hayır";
    if(c4 == 0)
        C4 = "Hayır";

    while (puan < 4)
    {
        cout<<"Lutfen soru seciniz: ";
        cin>>secim;

        if(secim=="1"){
            cout<<C1<<endl;
            puan++;
        }
        else if(secim=="2"){
            cout<<C2<<endl;
        }
        else if(secim=="3"){
            cout<<C3<<endl;
            puan++;
        }
        else if(secim=="4"){
            cout<<C4<<endl;
            puan++;
        }
        else
            cout<<"Hatalı secim.. "<<endl;
    }
}

void Cevap2(int c1, int c2, int c3, int c4)
```

```
{
    string C1 = "Evet",C2 = "Evet",C3 = "Evet",C4 = "Evet";
    int puan = 0;
    string secim;

    if(c1 == 0)
        C1 = "Hayır";
    if(c2 == 0)
        C2 = "Hayır";
    if(c3 == 0)
        C3 = "Hayır";
    if(c4 == 0)
        C4 = "Hayır";

    while (puan < 3)
    {
        cout<<"Lutfen soru seciniz: ";
        cin>>secim;

        if(secim == "1"){
            cout<<C1<<endl;
            puan++;
        }
        else if(secim == "2"){
            cout<<C2<<endl;

        }

        else if(secim == "3"){
            cout<<C3<<endl;
            puan++;

        }

        else if(secim == "4"){
            cout<<C4<<endl;
            puan++;

        }
        else
            cout<<"Hatalı seçim.."<<endl;
    }
}

void Cevap3(int c1, int c2, int c3, int c4)
{
    string C1 = "Evet",C2 = "Evet",C3 = "Evet",C4 = "Evet";
    int puan = 0;
    string secim;
```

```
if(c1 == 0)
    C1 = "Hayır";
if(c2 == 0)
    C2 = "Hayır";
if(c3 == 0)
    C3 = "Hayır";
if(c4 == 0)
    C4 = "Hayır";

while (puan < 3)
{
    cout<<"Lutfen soru seciniz: ";
    cin>>secim;

    if(secim == "1"){
        cout<<C1<<endl;
        puan++;
    }
    else if(secim == "2"){
        cout<<C2<<endl;

    }
    else if(secim == "3"){
        cout<<C3<<endl;
        puan++;
    }
    else if(secim == "4"){
        cout<<C4<<endl;
        puan++;
    }
    else
        cout<<"Hatalı seçim.."<<endl;
    }
}

void Cevap4(int c1, int c2, int c3, int c4)
{
    string C1 = "Evet",C2 = "Evet",C3 = "Evet",C4 = "Evet";
    int puan = 0;
    string secim;

    if(c1 == 0)
        C1 = "Hayır";
    if(c2 == 0)
        C2 = "Hayır";
    if(c3 == 0)
        C3 = "Hayır";
    if(c4 == 0)
        C4 = "Hayır";
```

```
while (puan < 4)
{
    cout<<"Lutfen soru seciniz: ";
    cin>>secim;

    if(secim == "1"){
        cout<<C1<<endl;
        puan++;
    }
    else if(secim == "2"){
        cout<<C2<<endl;
        puan++;
    }
    else if(secim == "3"){
        cout<<C3<<endl;
        puan++;
    }
    else if(secim == "4"){
        cout<<C4<<endl;
        puan++;
    }
    else
        cout<<"Hatalı giriş..."<<endl;
}
}

void Cevap5(int c1, int c2, int c3, int c4)
{
    string C1 = "Evet",C2 = "Evet",C3 = "Evet",C4 = "Evet";
    int puan = 0;
    string secim;

    if(c1 == 0)
        C1 = "Hayır";
    if(c2 == 0)
        C2 = "Hayır";
    if(c3 == 0)
        C3 = "Hayır";
    if(c4 == 0)
        C4 = "Hayır";

    while (puan < 3)
    {
        cout<<"Lutfen soru seciniz: ";
        cin>>secim;

        if(secim == "1"){
            cout<<C1<<endl;
            puan++;
        }
    }
}
```

```
else if(secim == "2"){
    cout<<C2<<endl;
    puan++;
}
else if(secim == "3"){
    cout<<C3<<endl;
    puan++;
}
else if(secim == "4"){
    cout<<C4<<endl;
}
else
    cout<<"Hatalı seçim.."<<endl;
}
}

void Case::Case_WriteToFile(){

    // Case_name_list -> ID ve Name
    ofstream Case_Write_ID_Name("Case_name_list.dat",ios::app);
    Case_Write_ID_Name<<ID<<" "<<Name<<endl;

    // XXX_story -> Story
    ofstream Case_Write_Story( Name+"_story.dat",ios::app);
    Case_Write_Story<<Story<<endl;

    // XXX_final -> Final
    ofstream Case_Write_Final( Name+"_final.dat",ios::app);
    Case_Write_Final<<Final<<endl;

    // XXX_person -> Person
    ofstream Case_Write_Person( Name+"_person.dat",ios::app);    // vakaya ait
    personun id ve name için bir txt dosyası oluşturuldu
    Case_Write_Person<<Person1.GetID()<<" "<<Person1.GetName()<<endl;

    // soru setlerinin story leri için txt dosyalarının oluşturulması
    ofstream Set1_Story( Name+"_question_set1_story.dat",ios::app);    // Case->Person-
    >Set1->story için txt oluşturuldu
    Set1_Story<<Person1.GetQuestionSet(1).GetStory()<<endl;

    ofstream Set2_Story( Name+"_question_set2_story.dat",ios::app);    // Case->Person-
    >Set2->story için txt oluşturuldu
    Set2_Story<<Person1.GetQuestionSet(2).GetStory()<<endl;

    ofstream Set3_Story( Name+"_question_set3_story.dat",ios::app);    // Case->Person-
    >Set3->story için txt oluşturuldu
    Set3_Story<<Person1.GetQuestionSet(3).GetStory()<<endl;
```

```
    ofstream Set4_Story( Name+"_question_set4_story.dat",ios::app);    // Case->Person-
>Set4->story için txt oluşturuldu
    Set4_Story<<Person1.GetQuestionSet(4).GetStory()<<endl;

    ofstream Set5_Story( Name+"_question_set5_story.dat",ios::app);    // Case->Person-
>Set5->story için txt oluşturuldu
    Set5_Story<<Person1.GetQuestionSet(5).GetStory()<<endl;

    //tüm soruların tek bir txt dosyasına yazılması
    ofstream all_question( Name+"_quesitons.dat",ios::app);    // Case->Person->Set5-
>story için txt oluşturuldu
    for(int set=1; set<6; set++){
        for(int q=1; q<5; q++){
            all_question<<Person1.GetQuestionSet(set).GetQuestion(q)<<endl;
        }
    }

    // dosyaya yazma : XXX_detective -> Person
    ofstream Case_Write_Detective( Name+"_detective.dat",ios::app);
    Case_Write_Detective<<Detective1.GetID()<<"_"<<Detective1.GetName()<<endl;    //
arada _ var istersen boşluk yapabilirsin sana bırakıyorum

}

void caseDuzenle(Case &Senaryo){
    string caseName;
    int caseID;

    cout<<"Senaryo ismi gir: ";
    cin>>caseName;
    cout<<"Senaryo ID gir: ";
    cin>>caseID;

    Senaryo.SetName(caseName);
    Senaryo.SetId(caseID);

    string caseStory;
    string caseFinal;

    cout<<"Senaryo hikayesi gir: ";
    cin>>caseStory;
    cout<<"Senaryo finali gir: ";
    cin>>caseFinal;

    Senaryo.SetStory(caseStory);
    Senaryo.SetFinal(caseFinal);

    string detectiveName;
```

```
int detectiveID;
cout<<"Dedektif ismi gir: ";
cin>>detectiveName;
cout<<"Dedektif ID gir: ";
cin>>detectiveID;

//Detektif eklendi
Detective other(detectiveID,detectiveName);
Senaryo.SetDetective(other);

//Questions story okuma

string questionStory1;
string questionStory2;
string questionStory3;
string questionStory4;
string questionStory5;

cout<<"Soru seti 1 için story gir";
cin>>questionStory1;
cout<<"Soru seti 2 için story gir";
cin>>questionStory2;
cout<<"Soru seti 3 için story gir";
cin>>questionStory3;
cout<<"Soru seti 4 için story gir";
cin>>questionStory4;
cout<<"Soru seti 5 için story gir";
cin>>questionStory5;

Questions set1;
Questions set2;
Questions set3;
Questions set4;
Questions set5;

//Soru seti id leri girildi
set1.SetID(1);
set2.SetID(2);
set3.SetID(3);
set4.SetID(4);
set5.SetID(5);

//soru seti storyleri girildi
set1.SetStory(questionStory1);
set2.SetStory(questionStory2);
set3.SetStory(questionStory3);
set4.SetStory(questionStory4);

//soru setlerine sorular girildi
```



```
string soru;
cout<<"Soru seti 1"<<endl;
for (int i = 1; i < 5; i++)
{
    cout<<"Soru " << i << " giriniz: ";
    cin>>soru;
    set1.SetQuestion(i,soru);
}

cout<<"Soru seti 2"<<endl;
for (int i = 1; i < 5; i++)
{
    cout<<"Soru " << i << " giriniz: ";
    cin>>soru;
    set2.SetQuestion(i,soru);
}

cout<<"Soru seti 3"<<endl;
for (int i = 1; i < 5; i++)
{
    cout<<"Soru " << i << " giriniz: ";
    cin>>soru;
    set3.SetQuestion(i,soru);
}

cout<<"Soru seti 4"<<endl;
for (int i = 1; i < 5; i++)
{
    cout<<"Soru " << i << " giriniz: ";
    cin>>soru;
    set4.SetQuestion(i,soru);
}

cout<<"Soru seti 5"<<endl;
for (int i = 1; i < 5; i++)
{
    cout<<"Soru " << i << " giriniz: ";
    cin>>soru;
    set5.SetQuestion(i,soru);
}

string personName;
int personID;
Person person;
cout<<"Person ID gir: ";
cin>>personID;
cout<<"Person Name gir: ";
cin>>personName;
```

```
//Person verileri girildi
person.SetID(personID);
person.SetName(personName);
//persona sorular eklendi
person.SetQuestionSet(1,set1);
person.SetQuestionSet(2,set2);
person.SetQuestionSet(3,set3);
person.SetQuestionSet(4,set4);
person.SetQuestionSet(5,set5);

// person eklendi
Senaryo.SetPerson(person);
}

#include "Detective.h"

Detective::Detective(){
    ID = 0;
    Name = "Default Detective";
}

Detective::Detective(int id, string name){
    Name = name;
    ID = id;
}

Detective::Detective(const Detective& other){
    Name = other.Name;
    ID = other.ID;
}

Detective& Detective::operator = (const Detective& other){
    Name = other.Name;
    ID = other.ID;
    return *this;
}

Detective::~~Detective(){}

void Detective::SetID(int id){
    ID = id;
}

void Detective::SetName(string name){
    Name = name;
}

int Detective::GetID(){
    return ID;
}
```

```
string Detective::GetName(){
    return Name;
}

ostream& operator << (ostream& os, const Detective detec){
    os<<"Name: "<<detec.Name<<endl<<"ID: "<<detec.ID;
    return os;
}

Detective operator + (Detective detec1, Detective detec2){
    string name = detec1.GetName() + detec2.GetName();
    int id = detec1.GetID() + detec2.GetID();
    Detective x(id,name);

    return x;
}

#include "File.h"

string Case_Read_ID_Name(){
    char ID;
    string x;
    ifstream dosyaOku("Case_name_list.dat");
    string satir = "";
    if ( dosyaOku.is_open() ){
        while ( getline(dosyaOku, satir) )
            cout << satir << endl;

        dosyaOku.close();
    }
    cout<<"Senaryo secmek icin ID giriniz: ";

    cin>>ID;
    ifstream dosyaOku2("Case_name_list.dat");
    satir = "";
    char id[100];

    if ( dosyaOku2.is_open() )
    {
        while ( getline(dosyaOku2, satir) ){
            strcpy(id, satir.c_str());

            if( id[0] == ID )
                x= satir;

        }
        dosyaOku2.close();
    }
    return x;
```

```
}

int getCaseID(string metin){
    string s = metin;
    string delimiter = "_";
    string token = s.substr(0,s.find(delimiter));

    return atoi(token.c_str());
}

string getCaseName(string metin){
    string delimiter = "_";
    string x = metin.erase(0, metin.find(delimiter) + delimiter.length());
    return x;
}

string Case_Read_Story(string name){    // Story bilgisini ekrana yazdıracak
    ifstream dosyaOku(name+"_story.dat");
    string satir = "";
    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);

    dosyaOku.close();
    return satir;
}

string Case_Read_Final(string name){
    ifstream dosyaOku(name+"_final.dat");
    string satir = "";
    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);

    dosyaOku.close();
    return satir;
}

string Case_Read_Detective(string name){
    ifstream dosyaOku(name+"_detective.dat");
    string satir = "";
    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);

    dosyaOku.close();
    return satir;
}

int getDetectiveID(string metin){
    string s = metin;
```

```
string delimiter = "_";
string token = s.substr(0,s.find(delimiter));

return atoi(token.c_str());
}

string getDetectiveName(string metin){
    string delimiter = "_";
    string x = metin.erase(0, metin.find(delimiter) + delimiter.length());
    return x;
}

string Case_Read_Person(string name){
    ifstream dosyaOku(name+"_person.dat");
    string satir = "";
    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);

    dosyaOku.close();
    return satir;
}

int getPersonID(string metin){
    char array[100];
    strcpy(array,metin.c_str());
    return array[0] - '0';
}

string getPersonName(string metin){
    char array[100];
    int i = 2;
    string name;
    strcpy(array,metin.c_str());

    while (array[i] != '\0')
    {
        name += array[i];
        i++;
    }

    return name;
}

string Case_Read_Set_Story(string name, int set_ID){

    string donder;
    if(set_ID == 1)
    {
```

```
    ifstream dosyaOku(name+"_question_set1_story.dat");
    string satir = "";

    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);
    donder = satir;
    dosyaOku.close();

}
else if(set_ID == 2)
{
    ifstream dosyaOku(name+"_question_set2_story.dat");
    string satir = "";

    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);
    donder = satir;
    dosyaOku.close();
}
else if (set_ID == 3)
{
    ifstream dosyaOku(name+"_question_set3_story.dat");
    string satir = "";

    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);
    donder = satir;
    dosyaOku.close();
}
else if (set_ID == 4)
{
    ifstream dosyaOku(name+"_question_set4_story.dat");
    string satir = "";

    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);
    donder = satir;
    dosyaOku.close();
}
else if(set_ID == 5)
{
    ifstream dosyaOku(name+"_question_set5_story.dat");
    string satir = "";

    if ( dosyaOku.is_open() )
        getline(dosyaOku, satir);
    donder = satir;
    dosyaOku.close();
}
else
```

```
        donder = "Hata";

    return donder;
}

string Case_Read_Quesitons(const string name, int q_ID){
    int k=1;
    ifstream dosyaOku;
    dosyaOku.open(name);
    string satir = "";
    string donder = "HATALI SECIM";
    if ( dosyaOku.is_open() ){
        while ( getline(dosyaOku, satir) ){
            if(k==q_ID){
                donder = satir;
            }

            k++;
        }
    }
    dosyaOku.close();
    return donder;
}

void readCase(Case &Senaryo){
    string metin = Case_Read_ID_Name();
    string caseName = getCaseName(metin);
    int id = getCaseID(metin);

    Senaryo.SetName(caseName);
    Senaryo.SetId(id);

    string caseStory = Case_Read_Story(caseName);
    string caseFinal = Case_Read_Final(caseName);

    Senaryo.SetStory(caseStory);
    Senaryo.SetFinal(caseFinal);

    string detectiveMetin = Case_Read_Detective(caseName);
    int detectiveID = getDetectiveID(detectiveMetin);
    string detectiveName = getDetectiveName(detectiveMetin);

    //Detektif eklendi
    Detective other(detectiveID,detectiveName);
    Senaryo.SetDetective(other);

    //Questions story okuma
    string questionStory1 = Case_Read_Set_Story(caseName,1);
    string questionStory2 = Case_Read_Set_Story(caseName,2);
    string questionStory3 = Case_Read_Set_Story(caseName,3);
```

```
string questionStory4 = Case_Read_Set_Story(caseName,4);
```

```
Questions set1;
```

```
Questions set2;
```

```
Questions set3;
```

```
Questions set4;
```

```
Questions set5;
```

```
//Soru seti id leri girildi
```

```
set1.SetID(1);
```

```
set2.SetID(2);
```

```
set3.SetID(3);
```

```
set4.SetID(4);
```

```
set5.SetID(5);
```

```
//soru seti storyleri girildi
```

```
set1.SetStory(questionStory1);
```

```
set2.SetStory(questionStory2);
```

```
set3.SetStory(questionStory3);
```

```
set4.SetStory(questionStory4);
```

```
//soru setlerine sorular girildi
```

```
int k = 1;
```

```
string adres = caseName + "_questions.dat";
```

```
ifstream dosyaOku(adres);
```

```
string satir;
```

```
if ( dosyaOku.is_open() ){
```

```
    while ( getline(dosyaOku, satir) ){
```

```
        if(k<=4)
```

```
            set1.SetQuestion(k,satir);
```

```
        else if(k<=8)
```

```
            set2.SetQuestion(k-4,satir);
```

```
        else if(k<=12)
```

```
            set2.SetQuestion(k-8,satir);
```

```
        else if(k<=16)
```

```
            set2.SetQuestion(k-12,satir);
```

```
        else if(k<=20)
```

```
            set2.SetQuestion(k-16,satir);
```

```
        cout<<satir<<endl;
```

```
        k++;
```

```
    }
```

```
}
```

```
dosyaOku.close();
```

```
string personMetin = Case_Read_Person(caseName);
```

```
string PersonName = getPersonName(personMetin);
```

```
int personID = getPersonID(personMetin);
```



```
Person person;

//Person verileri girildi
person.SetID(personID);
person.SetName(PersonName);
//persona sorular eklendi
person.SetQuestionSet(1,set1);
person.SetQuestionSet(2,set2);
person.SetQuestionSet(3,set3);
person.SetQuestionSet(4,set4);
person.SetQuestionSet(5,set5);

// person eklendi
Senaryo.SetPerson(person);
}

#include "File.h"

int main(){

    cout<<Case_Read_Story("Case");
}

#include "File.h"
#include "Exception.h"

int main(){

    Questions set1;
    Questions set2;
    Questions set3;
    Questions set4;
    Questions set5;

    set1.SetID(1);
    set1.SetStory("Olay bir deniz taşıtında yaşanmış olmalı sonuçta masa ve satranç tahtası
var. Acaba ne tür bir taşıt tekne olması çok olası bir durum...");
    set1.SetQuestion(1,"Üzerinde üniforma bulunan bu kişiler asker mi?");
    set1.SetQuestion(2,"Bu kişiler sizin askerleriniz miydi?");
    set1.SetQuestion(3,"Olay bir gemide mi yaşandı?");
    set1.SetQuestion(4,"Olay denizde mi yaşandı ?");

    set2.SetID(2);
    set2.SetStory("Bu kişiler bir başkası tarafından öldürülmemiş, amiralin ölümler ile bir
alakası olduğunu düşünmüştüm. Galiba yanlışım, tabi yalan söylemiyorsa...");
    set2.SetQuestion(1,"Olay vir deniz taşıtında mı yaşandı?");
    set2.SetQuestion(2,"Olay bir teknede mi yaşandı?");
    set2.SetQuestion(3,"Askerler birileri tarafından mı öldürüldü?");
```

```
set2.SetQuestion(4,"Bu kişiler dost muydu?");
```

```
set3.SetID(3);
```

```
set3.SetStory("Yaşlı amiral gözlerini odasında asılı duran silaha yöneltir. Bunlar silah ile mi öldü diye sorduğunda iki közünü de kapatır. Bu hem evet hem de hayır mı demek istedi, peki ama bunun anlamı ne?");
```

```
set3.SetQuestion(1,"Askerler birbirlerini mi vurdu?");
```

```
set3.SetQuestion(2,"Fotoğrafta gözükmeyen 3. bir şahıs var mıydı?");
```

```
set3.SetQuestion(3,"Ölmeden önce satranç mı oynadılar?");
```

```
set3.SetQuestion(4,"Ölümlerin oyun ile alakası var mı?");
```

```
set4.SetID(4);
```

```
set4.SetStory("Ortada bir satranç tahtası var ve birisi intihar etti acaba hangi intihar etti...");
```

```
set4.SetQuestion(1,"Askerlerden biri silah ile vurularak mı öldü ?");
```

```
set4.SetQuestion(2,"Silahla ölen kişi kaza sonucu mu öldü?");
```

```
set4.SetQuestion(3,"Silah ile ölen kişi intihar mı etti?");
```

```
set4.SetQuestion(4,"Deniz taşıtında olduğunu söylemiştin bu taşıt deniz altı mı?");
```

```
set5.SetID(5);
```

```
set5.SetStory("Askerler, satranç, intihar... taşlar şimdi yerine oturdu");
```

```
set5.SetQuestion(1,"Kazanan mı intihar etti?");
```

```
set5.SetQuestion(2,"Kaybeden boğularak mı öldü?");
```

```
set5.SetQuestion(3,"Deniz altı su mu alıyordu?");
```

```
set5.SetQuestion(4,"Bu bilgiler olayı çözmem için yeterli mi?");
```

```
Person amiral(1,"Amiral",set1,set2,set3,set4,set5);
```

```
Detective Orkun(1,"Orkun");
```

string story = "Orkun hafta sonunu değerlendirmek için sanat galerisine gider. Sanat galerisinde gezinirken bir süre sonra sıkılır ve çıkmaya karar verir ve çıkmadan önce son kez o fotoğrafa bakmak ister. Çok ilginç bir fotoğraf, tamamen siyah beyaz eski bir fotoğraf. Arka planı çok belli değildir ve mekânı kestirmek neredeyse imkansızdır. Fotoğrafi ilginç kılan şey ise karşılıklı oturan iki ceset yetmezmiş gibi bir de masanın üzerinde bir satranç tahtası olmasıdır. Bu fotoğrafın sırrını öğrenmek isteyen Orkun galeri görevlisi ile konuşur ve fotoğrafın izini sürmeye karar verir. Edindiği bilgiye göre bu fotoğraf galeriye emekli bir donanma amirali tarafından bağışlanmıştır. Uzun süren araştırmalar sonucunda bu amiral ile görüşme ayarlar. Görüşmeye giden Orkun hiç ummadığı bir şey ile karşılaşır. Amiral çok yaşlıdır ve konuşma yazma gibi motor fonksiyonlarını kaybetmiştir. Mektuplaştığı kişi yani amiralin eşi ona şunları söyler. "Biliyorum umduğunu bulamadın bende o fotoğrafın sırrını bilmiyorum, ne kadar konuşamasa da bazen soru sorduğumda gözleri ile evet ve hayır cevapları verebilir. Sağ gözünü kırparsa evet sol gözünü kırparsa hayır demek istiyordur." bunu üzerine Orkun evet ve hayırlı sorular sormaya başlar....";

string final = "Her şey anlaşıldı bu iki asker deniz altında görevli imiş, taşıtları hasar görmüş ve batıyormuş. Askerler öleceklerini anlayınca bir satranç oyunu oynamaya karar verirler. Kazanan kişi daha acısız ve kolay olan yolu yani silah ile intihar etmeye hak kazanmış. Kazanan kişi intihar eder kaybeden ise boğularak ölür. İyi ama diğeri neden silahla kendini vurmadı? Çünkü tek bir mermileri vardı. Bu sözler bizzat amiralin eşine aittir. Orkun döner ve amiralin eşine konuşur, ama bilmediğinizi söylemiştiniz der. Bu olayı çözen 5. kişinin der amiralin eşi. İnsan yaşlanınca geleni gideni azalıyor. Ben ve eşim böyle

bir oyun oynamaya karar verdik. Böylece son günlerimizde ziyaretçilerimiz çok oluyor bu saye de kendimizi yalnız hissetmiyoruz. Eşim yıllar önce çekti bu fotoğrafı. Olayın detaylarını ise boğulan askerin bıraktığı bir nottan öğrendik";

Case fotograf(1,"Fotograf",story,final,amiral,Orkun);

Case *olay = new Case;

Case oyun();

int secim;

bool flag = true;

fotograf.Case_WriteToFile();

do

{

try

{

cout<<"1. Yazar modu"<<

endl<<"2. Kayitli oyunlar"<<

endl<<"3. Yüklü oyun"<<

endl<<"4. Cikis"<<

endl<<"Secim: ";

cin>>secim;

switch (secim)

{

case 1:

caseDuzenle(*olay);

olay->Case_WriteToFile();

break;

case 2:

readCase(*olay);

olay->Play();

break;

case 3:

fotograf.Play();

break;

case 4:

flag = false;

break;

default:

throw Exception("seçim 1 ile 4 arasinda olmalı");

break;

}

}

catch(Exception obj)

{

cout<< obj.getMessage()<<endl<<endl;

cin.clear();

cin.ignore(100, '\n');

}// end catch

```
    } while (flag);

    return 0;
}

#include "Person.h"

Person::Person(){
    Questions setptr;
    ID = 0;
    Name = "Person";
    Set1 = setptr;
    Set2 = setptr;
    Set3 = setptr;
    Set4 = setptr;
    Set5 = setptr;
}

Person::Person(int id, string name, Questions set1, Questions set2, Questions set3,
Questions set4, Questions set5){
    ID = id;
    Name = name;
    Set1 = set1;
    Set2 = set2;
    Set3 = set3;
    Set4 = set4;
    Set5 = set5;
}

Person::Person(const Person& other){
    ID = other.ID;
    Name = other.Name;
    Set1 = other.Set1;
    Set2 = other.Set2;
    Set3 = other.Set3;
    Set4 = other.Set4;
    Set5 = other.Set5;
}

Person& Person::operator = (const Person& other){
    ID = other.ID;
    Name = other.Name;
    Set1 = other.Set1;
```

```
Set2 = other.Set2;
Set3 = other.Set3;
Set4 = other.Set4;
Set5 = other.Set5;
return *this;
}
Person::~~Person(){}

void Person::SetID(int id){
    ID = id;
}

void Person::SetName(string name){
    Name = name;
}

void Person::SetQuestionSet(int no, Questions& set){
    switch (no)
    {
        case 1:
            Set1 = set;
            break;
        case 2:
            Set2 = set;
            break;
        case 3:
            Set3 = set;
            break;
        case 4:
            Set4 = set;
            break;
        case 5:
            Set5 = set;
            break;
        default:
            cout<<"Hatalı numara girişı"<<endl;
            break;
    }
}

int Person::GetID(){
    return ID;
}

string Person::GetName(){
    return Name;
}

Questions& Person::GetQuestionSet(int no){
    switch (no)
```

```
{
    case 1:
        return Set1;
        break;
    case 2:
        return Set2;
        break;
    case 3:
        return Set3;
        break;
    case 4:
        return Set4;
        break;
    case 5:
        return Set5;
        break;
    default:
        cout<<endl<<"Hatalı getquestion no"<<endl;
        return Set1;
        break;
}
}

ostream& operator << (ostream& os, const Person person){
    os<<"ID: "<<person.ID<<
    endl<<"Name: "<<person.Name<<
    endl<<"set 1: "<<endl<<person.Set1<<
    endl<<"set 2: "<<endl<<person.Set2<<
    endl<<"set 3: "<<endl<<person.Set3<<
    endl<<"set 4: "<<endl<<person.Set4<<
    endl<<"set 5: "<<endl<<person.Set5<<endl;
    return os;
}

Person operator + (Person person1, Person person2){
    int id = person1.GetID() + person2.GetID();
    string name = person1.GetName() + person2.GetName();
    Questions SET1 = person1.GetQuestionSet(1) + person2.GetQuestionSet(1);
    Questions SET2 = person1.GetQuestionSet(2) + person2.GetQuestionSet(2);
    Questions SET3 = person1.GetQuestionSet(3) + person2.GetQuestionSet(3);
    Questions SET4 = person1.GetQuestionSet(4) + person2.GetQuestionSet(4);
    Questions SET5 = person1.GetQuestionSet(5) + person2.GetQuestionSet(5);

    Person x(id,name,SET1,SET2,SET3,SET4,SET5);

    return x;
}

#include "Questions.h"
```

```
Questions::Questions(){
    string BOS = "BOS";
    ID = 0;
    Story = Question1 = Question2 = Question3 = Question4 = BOS;
}

Questions::Questions(int id, string stry, string q1, string q2, string q3, string q4){
    ID = id;
    Story = stry;
    Question1 = q1;
    Question2 = q2;
    Question3 = q3;
    Question4 = q4;
}

Questions::Questions(const Questions& other){
    ID = other.ID;
    Story = other.Story;
    Question1 = other.Question1;
    Question2 = other.Question2;
    Question3 = other.Question3;
    Question4 = other.Question4;
}

Questions& Questions::operator = (const Questions& other){
    ID = other.ID;
    Story = other.Story;
    Question1 = other.Question1;
    Question2 = other.Question2;
    Question3 = other.Question3;
    Question4 = other.Question4;

    return *this;
}

Questions::~~Questions(){}

void Questions::SetID(int id){
    ID = id;
}

void Questions::SetStory(string stry){
    Story = stry;
}

void Questions::SetQuestion(int no, string q){
    switch (no)
    {

```

```
case 1:
    Question1 = q;
    break;
case 2:
    Question2 = q;
    break;
case 3:
    Question3 = q;
    break;
case 4:
    Question4 = q;
    break;
default:
    cout<<endl<<endl<<"Hatalı soru numarası girdiniz"<<endl<<endl;
    break;
}

}

int Questions::GetID(){
    return ID;
}

string Questions::GetStory(){
    return Story;
}

string Questions::GetQuestion(int no){
    switch (no)
    {
        case 1:
            return Question1;
            break;
        case 2:
            return Question2;
            break;
        case 3:
            return Question3;
            break;
        case 4:
            return Question4;
            break;
        default:
            return "Hatalı soru numarası girdiniz";
            break;
    }
}

ostream& operator << (ostream& os, const Questions que){
    os<<"ID: "<<que.ID<<
```



```
endl<<"Story: "<<que.Story<<
endl<<"Q1: "<<que.Question1<<
endl<<"Q2: "<<que.Question2<<
endl<<"Q3: "<<que.Question3<<
endl<<"Q4: "<<que.Question4<<endl;

return os;
}

Questions operator + (Questions q1, Questions q2){
    int id = q1.GetID() + q2.GetID();
    string stry = q1.GetStory() + q1.GetStory();
    string Q1 = q1.GetQuestion(1) + q1.GetQuestion(1);
    string Q2 = q1.GetQuestion(2) + q1.GetQuestion(2);
    string Q3 = q1.GetQuestion(3) + q1.GetQuestion(3);
    string Q4 = q1.GetQuestion(4) + q1.GetQuestion(4);
    Questions q(id,stry,Q1,Q2,Q3,Q4);
    return q;
}

#include "Test.h"

void TestDedective(){

    Detective d1;
    cout<<d1<<endl;

    Detective d2(1,"Ömer");
    cout<<d2<<endl;

    Detective d3(d2);
    cout<<d3<<endl;

    d1 = d2;
    cout<<d1<<endl;

    d1.SetID(100); d2.SetName("AAAAAA");

    cout<<d1.GetID()<<" "<<d1.GetName();
}

void TestQuestions(){
    Questions q1;
    cout<<q1<<endl;

    Questions q2(1,"hikaye","soru","soru","soru","soru");
    cout<<q2<<endl;

    Questions q3(q2);
```

```
cout<<q3<<endl;

q1 = q2;
cout<<q1<<endl;

q1 = q2 + q3;
cout<<q1<<endl;

q1.SetID(2);
q1.SetStory("aaaa");
q1.SetQuestion(1,"1");
q1.SetQuestion(2,"2");
q1.SetQuestion(3,"3");
q1.SetQuestion(4,"4");
cout<<q1<<endl;

cout<<q1.GetID()<<
endl<<q1.GetStory()<<
endl<<q1.GetQuestion(1)<<
endl<<q1.GetQuestion(2)<<
endl<<q1.GetQuestion(3)<<
endl<<q1.GetQuestion(4)<<endl;

}

void TestPerson(){
    Person p1;
    cout<<p1;
}
```