Main

```
using System;
using System.Text;

using OOD_Hospital_Project.AdapterPattern;
using OOD_Hospital_Project.SingletonPattern;
using OOD_Hospital_Project.BuilderPattern;
using OOD_Hospital_Project.Payments;
using OOD_Hospital_Project.Entrants;


namespace OOD_Proj_Final
{
    public partial class Form1 : Form
    {
        IEntrants Visitor1 = new Visitor("Paul S.", new DateTime(1999, 12, 23),
"+961 3 333444", new DateTime(2024, 6, 5));
        IEntrants Visitor2 = new Visitor("Tiana C.", new DateTime(2004, 3, 1), "+961
3 322508", new DateTime(2019, 3, 2));
        IEntrants VisitorWihoutBirthDate = new Visitor("Charbel k.", new
DateTime(2024, 5, 13));
        IPatient Patient1 = new InPatient("Jane BD.", new DateTime(1997, 9, 17),
"+961 1 223377", new DateTime(2020, 4, 5), true, new DateTime(2020, 4, 15));
        IPatient Patient2 = new InPatient("Jad K.", new DateTime(2011, 6, 8), "+961
7 228899", new DateTime(2019, 7, 9), false, new DateTime(2019, 8, 10));
        IPatient Patient3 = new OutPatient("Tony H.", new DateTime(2015, 3, 6),
"+961 3 765456", new DateTime(2024, 4, 4), 3);

        public Form1()
        {
            InitializeComponent();
            Patient1.SetPayment(PType.Cash, new DateTime(2002, 9, 8));
            Patient2.SetPayment(PType.WHISH, new DateTime(2019, 8, 5));
            Patient3.SetPayment(PType.OMT, new DateTime(2024, 4, 4));
            var HospitalInstance = Hospital.GetInstance();

            HospitalInstance.AddToList(Visitor1);
            HospitalInstance.AddToList(Visitor2);
            HospitalInstance.AddToList(VisitorWihoutBirthDate);
            HospitalInstance.AddToList(Patient1);
            HospitalInstance.AddToList(Patient2);
            HospitalInstance.AddToList(Patient3);

        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.Text = "PCT Hospital NDU Branch By Your Side!";

        }

        private void button1_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click_1(object sender, EventArgs e)
        {

        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            StringBuilder sb = new();
            sb.Append($"Name of the Hospital: {Hospital.GetInstance().HospitalName}
\n" +
                $"Hospital Location: {Hospital.GetInstance().HospitalLocation} \n" +
                $"Hospital Phone Number:
{Hospital.GetInstance().HospitalPhoneNumber}");
            MessageBox.Show(sb.ToString(), "Hospital Information");
        }

        private void button2_Click(object sender, EventArgs e)
        {

            SystemConnector systemConnector = new
HospitalManagementSystemConnector();
            Hospital.GetInstance().ExportData(systemConnector);
            string result =
Hospital.GetInstance().DisplayPatientRecords(systemConnector);
            MessageBox.Show(result, "Entrants Details");

        }

        private void button3_Click(object sender, EventArgs e)
        {

            HospitalRoomBuilder H1 = new RegularHospitalRoom();
            Hospital.GetInstance().Construct(H1);

            HospitalRoomBuilder H2 = new FirstClassHospitalRoom();
            Hospital.GetInstance().Construct(H2);

            string result = H1.Room.Display() +H2.Room.Display();
            MessageBox.Show(result, "Hospital Rooms");

        }

        private void button4_Click(object sender, EventArgs e)
        {
            StringBuilder PaymentResults = new();
            PaymentResults.Append(
             $"{Patient1.DisplayPatientPaymentInformation()}" +
              $"{Patient2.DisplayPatientPaymentInformation()}" +
             $"{Patient3.DisplayPatientPaymentInformation()}"
```

```
                            );
            MessageBox.Show(PaymentResults.ToString(), "Payment Details");
        }

        private void button5_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Hospital

```
using System;
using System.Text;

using OOD_Hospital_Project.AdapterPattern;
using OOD_Hospital_Project.BuilderPattern;
using OOD_Hospital_Project.Entrants;

namespace OOD_Hospital_Project.SingletonPattern
{
    public partial class Hospital
    {
        private static Hospital Instance;
        public string HospitalName { get; set; }
        public string HospitalLocation { get; set; }
        public string HospitalPhoneNumber { get; set; }
        public List<IEntrants> PatientsRecords { get; set; } = new
List<IEntrants>();
        List<IEntrants> Entrants { get; set; } = new List<IEntrants> { };


        private Hospital(string hospitalName, string hospitalLocation, string
hospitalNumber)
        {

            HospitalName = hospitalName;
            HospitalLocation = hospitalLocation;
            HospitalPhoneNumber = hospitalNumber;
        }

        public static Hospital GetInstance()
        {
            if (Instance == null)
            {
                Instance = new Hospital("PCT Medical Center", "Zouk Mosbeh", "+961 9
111222");
            }

            return Instance;
        }

        public void AddToList(IEntrants e)
        {
            Entrants.Add(e);
        }
        public void Construct(HospitalRoomBuilder RoomBuilder)
        {
            RoomBuilder.BuildBed();
            RoomBuilder.BuildArea();
            RoomBuilder.BuildView();
            RoomBuilder.BuildBathroom();
        }

        public void ExportData(SystemConnector Connector)
        {
            Connector.SendData(Entrants);
        }

        public string DisplayPatientRecords(SystemConnector Connector)
        {
            List<IEntrants> temp = Connector.GetData();
            StringBuilder recordBuilder = new StringBuilder();

            foreach (IEntrants entrant in temp)
            {
                string patientId = entrant is Patient patient ?
patient.PatientId.ToString() : "N/A";
                string entrantType = entrant switch
                {
                    IVisitor => "Visitor",
                    InPatient => "InPatient",
                    OutPatient => "OutPatient",
                    _ => "Unknown"
                };

                recordBuilder.Append($"Entrant Number: {patientId}\n" +
                                     $"{entrant.DisplayInfo()}\n");
            }

            return recordBuilder.ToString();

        }
    }
}
```

IEntrants

```
namespace OOD_Hospital_Project.Entrants
{
        public interface IEntrants
        {
        public string Name { get; set; }
        public DateTime? DateOfBirth { get; set; }
        public string PhoneNumber { get; set; }
        string DisplayInfo();
        }


}
```

IVisitor

```
      namespace OOD_Hospital_Project.Entrants
      {
          public interface IVisitor : IEntrants
          {
              DateTime VisitDate { get; set; }
              string DisplayInfo();
          }
}
```

Visitor

```
      namespace OOD_Hospital_Project.Entrants
      {
          public class Visitor : IVisitor
          {
              public string Name { get; set; }
              public DateTime? DateOfBirth { get; set; }
              public string PhoneNumber { get; set; }
              public DateTime VisitDate { get; set; }

              public Visitor(string n, DateTime DOB, string pn, DateTime vd)
              {
                  Name = n;
                  DateOfBirth = DOB;
                  PhoneNumber = pn;
                  VisitDate = vd;
              }

              public Visitor(string n, DateTime vd)
              {
                  Name = n;
                  DateOfBirth = null;
                  PhoneNumber = "N/A";
                  VisitDate = vd;
              }

              public string DisplayInfo()
              {
                  string dateOfBirthString = DateOfBirth.HasValue ? 
      DateOfBirth.Value.ToShortDateString() : "N/A";
                  return $"Entrant Type: Visitor\n" +
                         $"Name: {Name}\n" +
                         $"DateOfBirth: {dateOfBirthString}\n" +
                         $"PhoneNumber: {PhoneNumber}\n" +
                         $"Visit Date: {VisitDate.ToShortDateString()}\n";
              }
          }


                                             }
```

IPatient

```
using OOD_Hospital_Project.Payments;

namespace OOD_Hospital_Project.Entrants
{
    public interface IPatient : IEntrants
    {
        public int PatientId { get; set; }
        public Payment payment { get; set; }
        public void SetPayment(PType p, DateTime Dop);
        public string DisplayPatientPaymentInformation();
    }
}
```

Patient

```
using OOD_Hospital_Project.Payments;

namespace OOD_Hospital_Project.Entrants
{
    public class Patient : IPatient
    {
        public int PatientId { get; set; }
        public string Name { get; set; }
        public DateTime? DateOfBirth { get; set; }
        public string PhoneNumber { get; set; }
        public DateTime DateOfAdmission { get; set; }
        public Payment payment { get; set; }
        static int number = 1;

        public Patient(string name, DateTime dob, string pn, DateTime DA)
        {
            PatientId = number;
            Name = name;
            DateOfBirth = dob;
            PhoneNumber = pn;
            DateOfAdmission = DA;
            number++;
        }

        public virtual void SetPayment(PType p, DateTime Dop)
        {
        }

        public string DisplayPatientPaymentInformation()
        {
            string result = $"Patient Id: {PatientId}
\n{payment.DisplayPaymentInformation()}";
            return result;
        }

        public virtual string DisplayInfo()
        {
            string dateOfBirthString = DateOfBirth.HasValue ?
DateOfBirth.Value.ToShortDateString() : "N/A";
            return $"Name: {Name}\n" +
                $"DateOfBirth: {dateOfBirthString}\n" +
                $"PhoneNumber: {PhoneNumber}\n";        } } }
```

InPatient

```
using OOD_Hospital_Project.Payments;

namespace OOD_Hospital_Project.Entrants
{
    public class InPatient : Patient
    {
        public bool IsMonitored { get; set; }
        public int NumberOfStayInDays { get; set; }
        public DateTime CheckOutDate { get; set; }
        public InPatient(string name, DateTime dob, string pn, DateTime DA, bool Im,
DateTime checkOutDate) : base(name, dob, pn, DA)
        {
            IsMonitored = Im;
            CheckOutDate = checkOutDate;

            NumberOfStayInDays = (checkOutDate - DA).Days;
        }

        public override void SetPayment(PType p, DateTime Dop)
        {
            if (IsMonitored == true)
            {
                float amount = 100 * NumberOfStayInDays;
                payment = new(p, amount, Dop);
            }
            else {
                float amount = 80 * NumberOfStayInDays;
                payment = new(p, amount, Dop);
            }
        }

        public override string DisplayInfo()
        {
            return $"Entrant Type: InPatient\n" +
                   $"{base.DisplayInfo()}" +
                   $"Duration Of Stay: {NumberOfStayInDays} Days\n";
        }
    }


}
```

OutPatient

```
using OOD_Hospital_Project.Payments;

namespace OOD_Hospital_Project.Entrants
{
    public class OutPatient : Patient
    {
        public int NumberOfTests { get; set; }

        public OutPatient(string name, DateTime dob, string pn, DateTime DA, int
NOT) : base(name, dob, pn, DA)
```

```
        {
            NumberOfTests = NOT;
        }

        public override void SetPayment(PType p, DateTime Dop)
        {
            float amount = 25 * NumberOfTests;
            payment = new(p, amount, Dop);
        }

        public override string DisplayInfo()
        {
            return $"Entrant Type: OutPatient\n" +
                   $"{base.DisplayInfo()}" +
                   $"Numbers Of Tests to be Done: {NumberOfTests}\n";
        }

    }

}
```

PType

```
namespace OOD_Hospital_Project.Payments
{
    public enum PType
    {
        Cash, Visa, OMT, WHISH }  }
```

Payment

```
using System.Text;

namespace OOD_Hospital_Project.Payments
{
    public class Payment
    {
        IReceipt receipt;
        public PType PaymentType { get; set; }


        public Payment(PType p, float amount, DateTime Dop)
        {
            PaymentType = p;
            SetReceiptInformation(amount, Dop);

        }

        public void SetReceiptInformation(float amount, DateTime Dop)
        {
            receipt = new Receipt(amount, Dop);
        }

        public string DisplayPaymentInformation()
        {
            StringBuilder sb = new();
            string oldresult = receipt.DisplayReceipt();
            sb.Append($"{oldresult} \nThe Payment Type is {PaymentType}\n\n");
            return sb.ToString();
        }
    }
 }
```

IReceipt

```
namespace OOD_Hospital_Project.Payments
{
    public interface IReceipt
    {
        string DisplayReceipt();
    }
}
```

Receipt

```
using System.Text;
namespace OOD_Hospital_Project.Payments
{
    public class Receipt : IReceipt
    {
        public float Amount { get; set; }
        public DateTime DateOfPayment { get; set; }
        public Receipt(float amount, DateTime dateOfPayment)
        {
            Amount = amount;
            DateOfPayment = dateOfPayment;
        }

        public string DisplayReceipt()
        {
            StringBuilder sb = new StringBuilder();
            sb.Append($"The Total Amount is {Amount}$\n" +
                $"Date Of Payment: {DateOfPayment.ToShortDateString()}");
            return sb.ToString();
        }

    }

}
```

HospitalRoomBuilder

```
namespace OOD_Hospital_Project.BuilderPattern
{
    public abstract class HospitalRoomBuilder  {
        public HospitalRoom Room { get; set; }
        public abstract void BuildArea();
        public abstract void BuildView();
        public abstract void BuildBathroom();
        public abstract void BuildBed();

    }
}
```

RoomType

```
namespace OOD_Hospital_Project.BuilderPattern
{
    public enum RoomType
    {
        Regular, FirstClass
```

```
 }
    }
```

RegularHospitalRoom

```
namespace OOD_Hospital_Project.BuilderPattern
{
    public class RegularHospitalRoom : HospitalRoomBuilder
    {
        public RegularHospitalRoom()
        {
            Room = new HospitalRoom(RoomType.Regular);
        }
        public override void BuildArea()
        {
            Room.Area = 10;
        }
        public override void BuildView()
        {
            Room.View = "Regular View";
        }
        public override void BuildBathroom()
        {
            Room.Bathroom = "Simple Bathroom";
        }
        public override void BuildBed()
        {
            Room.Bed = 2;
        }
    }
}
```

FirstClassHospitalRoom

```
namespace OOD_Hospital_Project.BuilderPattern
{
    public class FirstClassHospitalRoom : HospitalRoomBuilder
    {
        public FirstClassHospitalRoom()
        {
            Room = new HospitalRoom(RoomType.FirstClass);
        }
        public override void BuildArea()
        {
            Room.Area = 20;
        }
        public override void BuildView()
        {
            Room.View = "Beautiful View";
        }
        public override void BuildBathroom()
        {
            Room.Bathroom = "Deluxe Bathroom";
        }
        public override void BuildBed()
        {
            Room.Bed = 1;
        }
    }

}
```

HospitalRoom

```
using System.Text;

namespace OOD_Hospital_Project.BuilderPattern
{
    public class HospitalRoom
    {
        private RoomType roomtype { get; set; }
        public int Bed { get; set; }
        public int Area { get; set; }
        public string Bathroom { get; set; }
        public string View { get; set; }

        public HospitalRoom(RoomType r)
        {
            roomtype = r;
        }

        public string Display()
        {
            StringBuilder sb = new StringBuilder();
            sb.Append($"The RoomType is {roomtype} \n" +
                $"The number of Bed is {Bed} \n" +
                $"The Area is {Area} m²\n" +
                $"The Bathroom is {Bathroom} \n" +
                $"The View is {View}\n\n");

            return sb.ToString();

        }
    }

}
```

SystemConnector

```
using OOD_Hospital_Project.Entrants;

namespace OOD_Hospital_Project.AdapterPattern
{
    public abstract class SystemConnector
    {
        public abstract void SendData(List<IEntrants> entrants);
        public abstract List<IEntrants> GetData();

    }
}
```

HospitalManagementSystemConnector

```
using OOD_Hospital_Project.Entrants;


namespace OOD_Hospital_Project.AdapterPattern
{
    public class HospitalManagementSystemConnector : SystemConnector
    {
        private HospitalManagementSystem HMS = new HospitalManagementSystem();
        public override void SendData(List<IEntrants> entrants)
        {

            HMS.SaveData(entrants);
        }

        public override List<IEntrants> GetData()
        {
            return HMS.GetData();
        }
    }

}
```

HospitalManagementSystem

```
using OOD_Hospital_Project.Entrants;

namespace OOD_Hospital_Project.AdapterPattern
{
    public class HospitalManagementSystem
    {
        public List<IEntrants> PatientRecords { get; set; }
        public void SaveData(List<IEntrants> entrants)
        {
            PatientRecords = entrants;
        }

        public List<IEntrants> GetData()
        {
            return PatientRecords;
        }
    }

 }
```