

```

1) public class Employee {
    public String Code { get; set; }
    public String Name { get; set; }
    public DateTime DateOf Birth { get; set; }
    public String Phone { get; set; }
    public String Address { get; set; }
    public int Salary { get; set; }

    public Employee() {}
    public Employee (string code, string name, DateTime date,
                    string phone, string address, int salary) {

        Code = code;
        Name = name;
        DateOf Birth = date;
        Phone = phone;
        Salary = salary;
    }
}

```

```

2. public class Employee {
    //...
    public override string ToString() {
        return $"{{Code}}; {{Name}}; {{DateOf Birth}}; {{Phone}}; {{Address}};
                {{Salary}}";
    }
}

```

3. public class Employee : Comparable<Employee> {

//...

public int CompareTo(Employee other) {

if (this.salary < other.salary)

return -1;

if (this.salary == other.salary)

return 0;

return 1;

}

}

4. public class Employee {

public String Name { get; set; }

public String Address { get; set; }

public List<Employee> EmployeeList { get; set; }

public Employee() {

EmployeeList = new List<Employee>();

}

public Employee(String name, String address,
List<Employee> list) {

Name = name;

Address = address;

EmployeeList = list;

}

}

5) a) // in Employee

```
public override bool Equals (object obj)
{
    if (obj == null)
        return false;

    if (!(obj is Employee))
        return false;

    Employee emp = obj as Employee;
    return Code == emp.Code && Name == emp.Name &&
        DateOf Birth == emp.DateOf Birth && Phone == emp.Phone;
}

public override int GetHashCode()
{
    return base.GetHashCode();
}
}
```

// in Employer

```
public void Add Employee (Employee e)
{
    bool isAdded = false;

    for each (var emp. in EmployeeList)
        if (emp.Equals(e)) {
            isAdded = true;
            break;
        }

    if (!isAdded)
        EmployeeList.Add(e);
}
```

5) &) // includes

using Newtonsoft.Json;

using System.IO;

// In App. config add <add key="Disk Path" value="D:\\" />

<appSettings>

<add key="Disk Path" value="D:\\" />

</appSettings>

// in class Employee

public void Save On Disk () {

string json = JsonConvert.SerializeObject(this);

string path = System.Configuration.ConfigurationManager.AppSettings["Disk Path"];

File.WriteAllText(\$"{path} \ {Name}.json", json);

}

6) // add <add key="Disk Path" value="D:\\" />

public delegate bool Func (Employee emp);

// in class Employee

public List <Employee> Filter (Func func)

{

return EmployeeList.Where (x => func(x)).ToList();

}

7) static void Main (string[] args)

{

List<Employee> list = new List<Employee>();

list.Add (new Employee ("C01", "Andrei P.", new DateTime (1980, 11, 30),
" ", "Giurgiu", 2200));

list.Add (new Employee ("C02", "Gortache G", new DateTime (1986, 10, 27),
"0720...", "Pitesti", 3600));

list.Add (new Employee ("C03", "Miron M.", new DateTime (2000, 8, 11),
"0730...", "Pitesti", 4000));

Employee emp = new Employee ("Miron D.", "Pitesti", list);

Func f1 = (Employee e) =>

{

int now = int.Parse (DateTime.Now.ToString ("yyyyMMdd"));

int dob = int.Parse (e.DateOfBirth.ToString ("... "));

int age = (now - dob) / 10000;

return e.Salary >= 2000 && age < 30;

};

Func f2 = (Employee e) =>

{

bool adr = e.Address.Split (';').Where (x => x.Contains ("Pitesti")).
ToList().Count > 0;

return adr && e.Phone.Length != 0;

};

// f1 -> C03 ; f2 -> C02, C03 ..