

Протокол №4

Тема:

Събитийно програмиране. Класове и методи.

Дата:

Изготвил:

### Въпроси:

1. Дефинирайте понятията: събитие, обработка на събитие и източник на събитие.
2. Дайте пример за събитие, като запишете програмен фрагмент и посочите кой е обработчика и кой е източника на събитието.

**Задача 1:** Създайте приложение, което да извършва следните стъпки:

Създайте един клас Person със свойства FirstName, LastName, Street, City, State, Zip, Email и Phone. Задайте на класа един метод GetAddress, който връща свойството на името и адреса на Person като низ във формата:

Alice Archer

100 Ash Ave

Bugsvilla CO 82010

1. Направете един нов клас Person чрез код, подобен на следния:

```
class Person{
    public string FirstName{ get;set;}
    public string LastName{ get;set;}
    // допишете и останалите свойства

    public string GetAddress() {
        return FirstName + " "+LastName+" "+"\n"+Street+"\n"+City+" "
        +State+" " +Zip;
    }
}
```

Наследете клас Employee от Person. Добавете свойствата EmployeeId и MailStop.

1. Направете класът Employee да бъде подобен на следния:

```
class Employee: Person
{
    public int EmployeeId { get;set;}
    public string MailStop { get; set; }
}
```

Наследете клас Manager от Employee. Добавете свойството DepartmentName и свойството DirectReports от типа List< Employee >. Създайте метод GetDirectReportsList, който връща имената на Employees на Manager, разделени чрез нови редове.

1. Направете класът Manager да бъде подобен на следния:

```
class Manager:Employee
{
    public string DepartmentName { get; set; }
    public List<Employee> DirectReports = new List<Employee>();
    public string GetDirectReportsList()
    {
```

```

        string result = “ ”;
        foreach( Employee emp in DirectReports){
            result +=emp.FirstName + “ ”+emp.LastName + “\n”;
        }
        return result;
    }
}

```

Накарайте главната програма да създава два Employee с имена Alice и Bob, един Manager с име Cindy, който има Alice и Bob в своя отдел, и Person с име Dan.

1. Тъй като бутоните на програмата трябва да достъпва обектите, тези обекти трябва да бъдат записани в полета от ниво клас, както е показано в следващия код:
 

```

private Person Dan;
private Employee Alice, Bob;
private Manager Cindy;

```
2. Добавете към обработчика на събитието Load на главната форма код, който да инициализира обектите. Следващият код показва как програмата може да създаде обект Employee за Alice.
 

```

Alice.City = “Bugsville”;
Alice.State = “CO”;
Alice.Zip = “82010”;
Alice.EmployeeId = “1001”;
Alice.MailStop = “A-1”;

```
3. Създаването и инициализирането на другите обекти става по подобен начин. Единственото необикновено нещо е добавянето на Alice и Bob като служители на Cindy, както е показано в следващия код:
 

```

Cindy.DirectReports.Add(Alice);
Cindy.DirectReports.Add(Bob);

```

Направете един метод ShowAddress, който приема Person като параметър и показва адреса на Person.

1. Използвайте код, подобен на следния

```

private void ShowAddress( Person person) {
    MessageBox.Show(person.GetAddress());
}

```

Върху главната форма създайте бутони, които извикват ShowAddress за всеки човек, подавайки на метода правилния обект.

1. Създайте обработчици на събития на бутона Click. Следващият код показва обработчика на събитие, който показва адреса на Cindy:

```

private void cindyAddressButton_Click(object sender, EventArgs e) {
    ShowAddress(Cindy);
}

```

Създайте един завършващ бутон, който показва списък от директно рапортуващи на Cindy.

1. Този метод просто извикваме метода `GetDirectReportsList` на обекта `Cindy` и показва резултата:

```
private void cindyReportsButton_Click(object sender, EventArgs e) {  
    MessageBox.Show(Cindy.GetDirectReportsList());  
}
```