

---

---

# حل مسائل C#

## (مرجع کامل)

---

---

تالیف

مهندس رمضان عباس نژادورزی



فن آوری نوین

---

---

|                     |  |
|---------------------|--|
| سرشناسه             | : عباس نژاد، رمضان، ۱۳۴۸ -                                       |
| عنوان و نام پدیدآور | : حل مسائل C# (مرجع کامل) / رمضان عباس نژادورزی.                 |
| مشخصات نشر          | : بابل: فناوری نوین، ۱۳۸۸.                                       |
| مشخصات ظاهری        | : 208ص.: مصور، جدول.   |
| شابک                | : 6200 ریال: 9786009141340                                       |
| وضعیت فهرست نویسی   | : فیپا   |
| موضوع               | : سی شارپ (زبان برنامه نویسی کامپیوتر)                           |
| موضوع               | : سی شارپ (زبان برنامه نویسی کامپیوتر) -- مسائل، تمرین ها و غیره |
| رده بندی کنگره      | : ۱۳۸۸ س ۹۵ع ۷۶/۲۳ QA۷۳  |
| رده بندی دیویی      | : ۱۳۳/۰۰۵  |
| شماره کتابشناسی ملی | : ۱۹۵۸۱۹۶  |



فن آوری نوین

[www.fanavarinovin.net](http://www.fanavarinovin.net)

بابل، کد پستی 47167-73448

تلفن: 011-32256687

## حل مسائل C# (مرجع کامل)

تألیف: مهندس رمضان عباس نژادورزی

ناشر: فن آوری نوین

چاپ اول: زمستان 1388

تیراژ: 1000 جلد

طراح جلد: احمد فرجی

شابک: 978 - 600 - 91413 - 4 - 0

حروفچینی و صفحه آرای: فن آوری نوین

قیمت: 6200 تومان

66400144-66400220

تهران، خ اردیبهشت، نبش وحید نظری، پلاک 142

تلفکس:

## فهرست مطالب

|                                    |     |
|------------------------------------|-----|
| فصل اول: آشنایی با زبان C#         | 7   |
| فصل دوم: ساختار تصمیم و حلقه تکرار | 15  |
| فصل سوم: آرایه‌ها و رشته‌ها        | 52  |
| فصل چهارم: کلاس‌ها و اشیا          | 91  |
| فصل پنجم: مباحث پیشرفته در C#      | 145 |
| پیوست: پروژه‌های حل شده در سایت    | 205 |
| منابع                              | 207 |



## مقدمه

زبان C# در فناوری دات نت (NET) توسط شرکت مایکروسافت ارائه شده است که کاملاً شیء‌گرا است. امروزه، اکثر دانشجویان رشته کامپیوتر و فناوری اطلاعات با این زبان آشنایی دارند. زیرا، این زبان از محبوب‌ترین زبان‌های خانواده C می‌باشد. از طرف دیگر، زبان C# به عنوان یکی از مهم‌ترین زبان‌های رشته کامپیوتر، فناوری اطلاعات، ICT، علوم کامپیوتر و رشته‌های دیگر تحت عنوان درس زبان برنامه‌نویسی پیشرفته تدریس می‌شود. یکی از روش‌های موفق یادگیری زبان‌های برنامه‌نویسی، حل مسائل گوناگون می‌باشد.

در حال حاضر کتاب‌های متعددی برای زبان برنامه‌نویسی C# ارائه شده است که جای تقدیر و تشکر دارد. اما، این کتاب با بهره‌گیری از سال‌ها تجربه در زمینه تدریس، تالیف، برنامه‌نویسی و مدیریت پروژه‌های نرم‌افزاری تدوین شده است. از ویژگی‌های بارز این کتاب، بیان مثال‌های کاربردی و حل گام به گام به همراه توضیحات دقیق آن‌ها است.

مثال‌هایی که در کتاب آورده شده‌اند، همگی دارای کاربرد و هدف خاصی در برنامه‌نویسی هستند و برخی از آن‌ها را می‌توانید در پروژه‌های کاربردی و بزرگ استفاده کنید.

در این کتاب به آموزش بانک اطلاعاتی پرداختیم. زیرا، مطالب آن گسترده است و بررسی برنامه‌نویسی بانک اطلاعاتی در این کتاب نمی‌گنجد. بنابراین، اگر می‌خواهید از زبان C# برای برنامه‌نویسی بانک اطلاعاتی استفاده کنید، به کتاب آموزش گام به گام برنامه‌نویسی بانک اطلاعاتی با C# نوشته همین مولف از انتشارات فن‌آوری نوین مراجعه نمایید.

امیدوارم این کتاب مورد قبول اساتید، دانشجویان و خوانندگان محترم واقع شود. کتاب شامل برنامه‌هایی است که کد آن‌ها را می‌توانید از سایت انتشارات فن‌آوری نوین به آدرس [www.fanavarinovin.net](http://www.fanavarinovin.net) بگیرید.

رمضان عباس‌نژاد ورزی

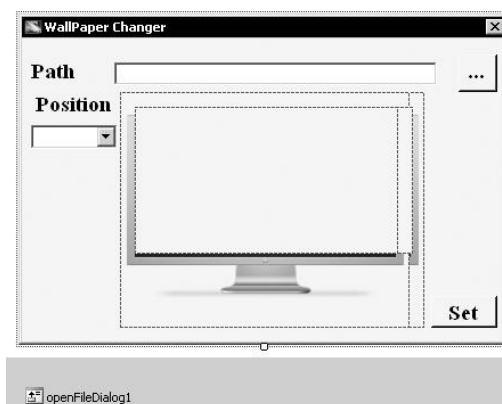
[fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com)

# نمونه برنامه‌های حل شده

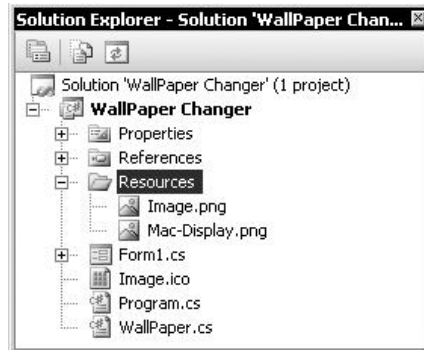
برنامه‌ای که امکان تغییر تصویر دسک‌تاپ کامپیوتر را فراهم می‌کند.

مراحل طراحی و اجرا:

1. پروژه جدیدی به نام Wallpaper\_Changer ایجاد کنید و خاصیت Text فرم را عبارت Wallpaper Changer تعیین نمایید.
2. دو کنترل Label به فرم اضافه کنید و خاصیت Text آن‌ها را Path و Position تعیین نمایید.
3. دو کنترل Button به فرم اضافه کرد، خاصیت Text آن‌ها را Set و ... تعیین نمایید.
4. یک کنترل TextBox به فرم اضافه کنید.
5. چهار کنترل PictureBox به فرم اضافه کرده، Image آن‌ها را با تصویر روی فرم تغییر دهید.
6. یک کنترل ComboBox به فرم اضافه کرده، در خاصیت Items آن عبارت Title, Center, Stretch را اضافه کنید.



7. یک کنترل openFileDialog به فرم اضافه کنید.
8. در پنجره Solution Explorer، دو تصویر زیر را در آیکن (پوشه) Resources اضافه کنید:



9. گزینه View/Code را اجرا کرده و System.Runtime.InteropServices را با دستور زیر به برنامه اضافه کنید:

```
using System.Runtime.InteropServices;
```

10. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1\_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    Wallpaper.Set(Application.StartupPath+"\\Test.jpg", Wallpaper.Style.Stretched;
}
```

11. دکمه Set را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text != "")
    {
        Wallpaper.Style style = new Wallpaper.Style();
        switch (comboBox1.Text)
        {
            case "Stretch":
                style = Wallpaper.Style.Stretched;
                break;
            case "Tile":
                style = Wallpaper.Style.Tiled;
                break;
            case "Center":
                style = Wallpaper.Style.Centered;
                break;
        }
        Wallpaper.Set(textBox1.Text, style);
    }
}
```

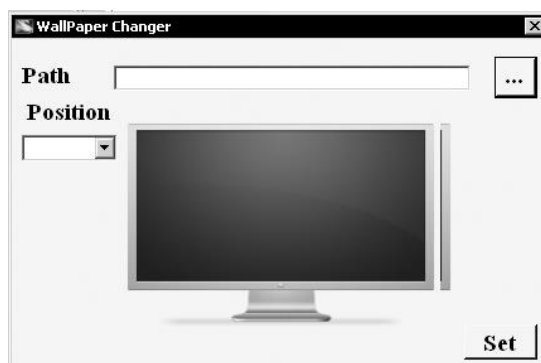
این دستورات، ابتدا با توجه به انتخاب کاربر از طریق comboBox سبک نمایش (Wallpaper.Style) را انتخاب کرده، تصویر که نام آن در textBox1 باشد را بروی دسکتاپ کامپیوتر اعمال می‌کند (با اجرای Set بروی Wallpaper).

**12.** دکمه ... را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

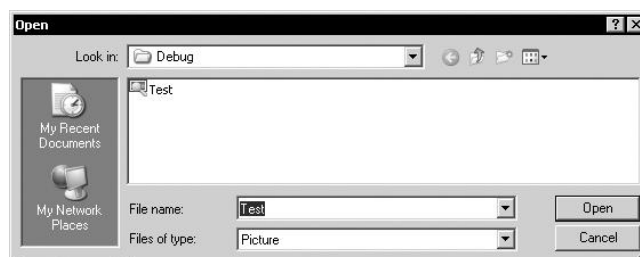
```
private void button2_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        textBox1.Text = openFileDialog1.FileName;
        pictureBox2.Image = Image.FromFile(openFileDialog1.FileName);
        pictureBox2.Visible = true;
        comboBox1.SelectedIndex = 2;
    }
}
```

کاربر با این دستورات می‌تواند تصویری را از طریق openFileDialog1 به pictureBox2 انتقال دهد.

**13.** پروژه را ذخیره و اجرا کنید تا تصویر زیر ظاهر شود:



دکمه ... را کلیک کرده تا پنجره Open ظاهر شود (شکل زیر):



در این پنجره تصویر Test را کلیک مضاعف کرده تا شکل فرم به صورت زیر تغییر کند:





دکمه Set را کلیک کنید تا تصویر انتخاب شده بر روی دسکتاپ کامپیوترتان قرار گیرد (شکل زیر):



کلاسی به نام Date بنویسید که دارای ویژگی‌های زیر باشد:

- ❖ سال، ماه و روز را بپذیرد.
- ❖ سازنده‌ای بنویسید که اگر ماه بین 1 تا 6 است، کاربر نتواند روز را بیشتر از 31 وارد نماید.
- ❖ ولی، اگر ماه بین 7 تا 11 است. کاربر نتواند روز را بیشتر از 30 وارد نماید و اگر ماه 12 بود، کاربر نتواند روز را بیشتر از 29 وارد کند. در این سازنده از ورود سال، ماه و روز منفی جلوگیری می‌کند.
- ❖ متد Equals را مجدداً پیاده‌سازی کنید تا دو تاریخ را با یکدیگر مقایسه کند.
- ❖ متدی به نام nextDay بنویسید که تاریخ را یک روز اضافه کند.
- ❖ متدی بنویسید که حاصل جمع دو تاریخ را برگرداند.
- ❖ عملگر – مجدداً تعریف کنید تا دو تاریخ را از هم تفریق کند.

❖ متد ToString() را مجدداً پیاده‌سازی کنید تا تاریخ را به فرمت روز/ماه/سال تبدیل کرده برگرداند(سال چهارم رقم، ماه دوم رقم و روز دوم رقم).

مراحل طراحی و اجرا

1. پروژه جدیدی به نام Date ایجاد کنید.

2. گزینه Project /Add Class را اجرا کنید. در کادری که ظاهر می‌شود، نام کلاس را Date.cs انتخاب کرده تا کلاس Date به پروژه اضافه شود. اکنون دستورات آن را به صورت زیر تغییر دهید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Date
{
    class Date
    {
        private int day;
        private int month;
        private int year;
        public Date(int y, int m, int d)
        {
            setYear(y);
            setMonth(m);
            setDay(d);
        }
        public int Day
        {
            get { return day; }
            set
            {
                setDay(value);
            }
        }
        public int Month
        {
            get { return month; }
            set
            {
                setMonth(value);
            }
        }
        public int Year
        {
            get { return year; }
            set
            {
                setYear(value);
            }
        }
        private void setDay(int d)
        {
            if (month > 0 && month < 7)
                if (d > 0 && d <= 31) day = d;
                else day = 0;
            else if (month >= 7 && month < 12)
                if (d > 0 && d <= 30) day = d;
                else day = 0;
            else if (month == 12)
```

```

        if (d > 0 && d <= 29) day = d;
        else day = 0;

    }
    private void setMonth(int m)
    {
        if (m > 0 && m <= 12)
            month = m;
        else month = 0;
    }
    private void setYear(int y)
    {
        if (y > 0) year = y;
        else year = 0;
    }
    public bool Equals(Date d1)
    {
        return (this.day == d1.day && this.month == d1.month &&
            this.year == d1.year);
    }
    public Date nextDay()
    {
        Date d1= new Date(0,0,0);
        d1 = this;
        d1.day++;
        if (d1.day > 31 && d1.month <= 6)
        {
            d1.month ++;
            d1.day = 1;
        }
        else if (d1.day>30 && d1.month>6 && d1.month <= 11)
        {
            d1.month ++;
            d1.day = 1;
        }
        else if (d1.day > 29 && d1.month == 12)
        {
            d1.month ++;
            d1.day = 1;
        }
        if (d1.month > 12)
        {
            d1.year++;
            d1.month = 1;
        }
        return d1;
    }
    public Date addDate(Date d)
    {
        Date d1 = new Date(0, 0, 0);
        d1.day = (int) this.day + d.day ;
        d1.month = (int)this.month + d.month;
        d1.year = (int)this.year + d.year;
        if (d1.day > 31 && d1.month <= 6)
        {
            d1.month++;
            if (d1.month == 7)
                d1.day = 30 - d1.day;
            else
                d1.day = 31 - d1.day;
        }
        return d1;
    }

```

```

    }
    if (d1.day > 29 && d1.month == 12)
    {
        d1.month++;
        d1.day = 29 - d1.day;
        if (d1.month > 12)
        {
            d1.month = 1;
            d1.year++;
        }
        return d1;
    }
    if (d1.month > 12 )
    {
        if (d1.day > 31)
        {
            d1.month = (d1.month + 1) % 12;
            d1.day = d1.day - 29;
        }
        else d1.month = d1.month % 12;
        if (d1.month == 0) d1.month++;
        d1.year = d1.year + (int) d1.month / 12;
        return d1;
    }
    return d1;
}

public override string ToString()
{
    string res = "";
    res = this.year.ToString() + "/";
    if (this.month < 10)
        res = res + "0" + this.month.ToString() + "/";
    else
        res = res + this.month.ToString() + "/";
    if (this.day < 10)
        res = res + "0" + this.day.ToString();
    else
        res = res + this.day.ToString();
    return res;
}

public static Date operator -(Date d1, Date d2)
{
    Date d3 = new Date(0, 0, 0);
    if (d1.day < d2.day )
    {
        if (d1.month <= 6) d1.day = d1.day + 31;
        else if (d1.month <= 11) d1.day = d1.day + 30;
        else d1.day = d1.day + 29;
        d1.month = d1.month - 1;
    }
    if (d1.month < d2.month )
    {
        d1.month = d1.month + 12;
        d1.year = d1.year - 1;
    }
    d3.day = d1.day - d2.day;
    d3.month = d1.month - d2.month;
    d3.year = d1.year - d2.year;
    return d3;
}

public bool chekDate()

```

```

    {
        return (!(this.day == 0 || this.month == 0 ||
            this.year == 0));
    }
}

```

این دستورات کلاس Date را تعریف می کنند که دارای اعضای زیر می باشد:

- ❖ **فیلدهای day, month و year**: به ترتیب برای نگهداری روز، ماه و سال به کار می روند.
- ❖ **خواص Day, Month و Year**: به ترتیب مقادیر فیلدهای day, month و year را بازیابی کرده یا مقداردهی می کنند.

❖ **سازنده Date**: سه پارامتر y, m و d را گرفته با فراخوانی توابع setYear, setMonth و setDay آنها را در فیلدهای year, month و day قرار می دهد.

❖ **متد setDay**: پارامتر d (روز) را گرفته، اگر ماه بین 1 تا 6 باشد و روز بین 1 تا 31 باشد، d را در day قرار می دهد، وگرنه day را برابر صفر قرار می دهد. اگر ماه بین 1 تا 6 نباشد، اگر ماه بین 7 تا 11 باشد و روز بین 1 تا 30 باشد، روز (d) را در day قرار می دهد. اگر ماه برابر 12 باشد و روز بین 1 تا 29 باشد، d در day قرار می گیرد، وگرنه، day برابر صفر می شود. این دستورات، روز (day) را مقدار می دهند.

❖ **متد setMonth**: پارامتر m (ماه) را گرفته، اگر در بین 1 تا 12 باشد در فیلد month قرار می دهد. در غیر این صورت، 0 را در month قرار می دهد.

❖ **متد setYear**: پارامتر y را گرفته در فیلد year قرار می دهد.

❖ **متد Equals**: دو تاریخ را مقایسه می کند. یکی از این تاریخ ها را به صورت ضمنی و دیگری را با پارامتر d1 دریافت می نماید.

❖ **متد nextDay**: تاریخ را یک روز جلو می برد. برای این منظور یک تاریخ را به صورت ضمنی گرفته و یک نمونه جدید به نام d1 ایجاد می کند. تاریخ که به صورت ضمنی گرفته (this) را در d1 قرار می دهد. در ادامه، فیلد day، d1 را یک واحد اضافه می کند. اگر d1.day بزرگتر از 31 باشد و month کوچکتر یا مساوی 6 باشد، month مربوط به d1 را یک واحد اضافه می کند و day مربوط به d1 را برابر 1 قرار می دهد. وگرنه، اگر روز بزرگتر از 30 و ماه بین 6 تا 11 باشد، ماه را یک واحد اضافه می کند و روز را برابر با یک قرار می دهد. وگرنه، اگر ماه برابر با 12 و روز بزرگتر از 29 باشد، ماه را یک واحد اضافه می کند. و روز را برابر با یک قرار می دهد. در پایان، اگر ماه بزرگتر از 12 باشد، به سال یک واحد اضافه می نماید و ماه برابر با یک خواهد شد.

❖ **متد addDate**: یک پارامتر را به طور ضمنی گرفته آن را با پارامتر d جمع کرده در d1 قرار می دهد.

❖ **متد ToString**: یک تاریخ را به طور ضمنی گرفته آن را به رشته تبدیل می کند و بر می گرداند.

❖ **متد operator**: عملگر - را مجددا پیاده سازی می کند که پارامتر d1 و d2 را گرفته مقدار d1 - d2 را برمی گرداند.

❖ **متد chekDate**: تاریخی را به طور ضمنی گرفته آن را چک می کند. اگر روز، ماه یا سال مخالف صفر

باشد، true، وگرنه، false را برمی گرداند.

### 3. سه کنترل Label، سه کنترل maskedTextBox

و پنج کنترل Button به فرم اضافه کنید.

### 4. ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1\_Load را به صورت زیر تغییر دهید:

```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "Date 1";
    label2.Text = "Date 2";
    label3.Text = "Result";
    maskedTextBox1.Mask = "0000/00/00";
    maskedTextBox2.Mask = "0000/00/00";
    maskedTextBox3.ReadOnly = true;
    button1.Text = "Add";
    button2.Text = "Sub";
    button3.Text = "Next Day";
    button4.Text = "Equals";
    button5.Text = "Exit";
}
```

این دستورات خواص کنترل‌های روی فرم را مقداردهی می‌کنند.

### 5. به قبل از رویداد Form1\_Load بروید و دستور زیر را تایپ کنید:

```
Date d1, d2, d3;
```

این دستور سه نمونه d1، d2 و d3 را از نوع Date تعریف می‌کند.

### 6. متد readDate، را بعد از تعریف متغیرها به صورت زیر معرفی کنید:

```
private bool readDate()
{
    d1 = new Date(Convert.ToInt16(maskedTextBox1.Text.Substring(0, 4)), Convert.ToInt16(maskedTextBox1.Text.Substring(5, 2)), Convert.ToInt16(maskedTextBox1.Text.Substring(8, 2)));
    if (d1.chekDate() == false)
    {
        MessageBox.Show("Please enter correct date1");
        return false;
    }
    d2 = new Date(Convert.ToInt16(maskedTextBox2.Text.Substring(0, 4)), Convert.ToInt16(maskedTextBox2.Text.Substring(5, 2)), Convert.ToInt16(maskedTextBox2.Text.Substring(8, 2)));
    if (d2.chekDate() == false)
    {
        MessageBox.Show("Please enter correct date1");
        return false;
    }
    return true;
}
```

این دستورات تاریخ‌ها را از طریق کنترل‌های maskedTextBox1 و maskedTextBox2 خوانده در نمونه‌های d1 و d2 قرار می‌دهند. اگر تاریخ درست وارد نشود، این متد false را برمی‌گرداند.

7. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    if (readDate())
    {
        d3 = d1.addDate(d2);
        maskedTextBox3.Text = d3.ToString();
    }
}
```

این دستورات، با فراخوانی متد readDate دو تاریخ را می‌خواند. اگر دو تاریخ صحیح وارد شوند، تاریخ‌های d1 و d2 را با فراخوانی متد addDate جمع کرده در d3 قرار می‌دهد و سپس، d3 را به رشته تبدیل می‌کند (با فراخوانی ToString) و آن را در maskedTextBox3 نمایش می‌دهد.

8. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    if (readDate())
    {
        d3 = d1 - d2;
        maskedTextBox3.Text = d3.ToString();
    }
}
```

این دستورات، با فراخوانی متد readDate دو تاریخ را خوانده اگر تاریخ‌ها درست وارد شده باشند، آن‌ها را در d1 و d2 قرار داده سپس با عملگر -، مقدار d1 را از d2 کم کرده در d3 قرار می‌دهد و d3 را نمایش می‌دهد.

9. دکمه button3 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button3_Click(object sender, EventArgs e)
{
    d1 = new Date(Convert.ToInt16(maskedTextBox1.Text.Substring(0, 4)),
        Convert.ToInt16(maskedTextBox1.Text.Substring(5, 2)),
        Convert.ToInt16(maskedTextBox1.Text.Substring(8, 2)));
    if (d1.chekDate() == false)
    {
        MessageBox.Show("Please enter correct date1");
        return;
    }
    d3 = d1.nextDay();
    maskedTextBox3.Text = d3.ToString();
}
```

این دستورات، یک تاریخ را خوانده در d1 قرار می‌دهد. سپس، با فراخوانی متد nextDay، تاریخ خوانده شده را یک روز جلو برده در d3 قرار می‌دهد و d3 را نمایش می‌دهد.

**10.** دکمه button4 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button4_Click(object sender, EventArgs e)
{
    if (readDate())
    {
        if (d1.Equals(d2)) MessageBox.Show("Equals");
        else MessageBox.Show("Not Equals"); ;
    }
}
```

این دستورات، با فراخوانی متد readDate دو تاریخ را خوانده اگر دو تاریخ برابر باشند ( نتیجه متد Equals برابر true باشد)، پیغام Equals، وگرنه پیغام Not Equals را نمایش می دهند.

**11.** دکمه button5 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button5_Click(object sender, EventArgs e)
{
    Close();
}
```

**12.** پروژه را ذخیره و اجرا کنید. در جلوی Date1 و Date2 به ترتیب مقادیر 1387/01/02 و 1380/05/08 را وارد کرده، دکمه Add را کلیک کنید تا خروجی را به شکل زیر ببینید:

اکنون دکمه sub را کلیک کنید تا تفاضل دو تاریخ را مشاهده کنید( شکل زیر):

در maskedTextBox3 اختلاف دو تاریخ را می بینید. اکنون دکمه Next Day را کلیک نمایید تا تاریخ روز بعد تاریخی که در maskedTextBox1 وارد شده است را در maskedTextBox3 ببینید (شکل زیر):



دکمه Equals را کلیک کنید تا خروجی مقابل را مشاهده نمایید.

دکمه Exit را کلیک کنید تا از برنامه خارج شوید.

برنامه‌ای که دو آرایه  $n$  عنصری ( $n$  را از ورودی می‌خواند) تعریف کرده توسط تابع تصادفی اعداد بین 0 تا 99 تولید کرده در این دو آرایه قرار می‌دهد. سپس، اعمال زیر را انجام می‌دهد:

- ❖ متدی عناصر مشترک دو آرایه را پیدا می‌کند.
- ❖ متدی اجتماع دو آرایه را پیدا می‌کند.
- ❖ متدی تفاضل دو آرایه را برمی‌گرداند (توسط یک CheckBox تعیین می‌گردد که کدام آرایه از آرایه دوم کم گردد).
- ❖ برای این که تعیین شود کدام یک از اعمال انجام گردد، از سه RadioButton استفاده می‌شود.

مراحل طراحی و اجرا

1. پروژه جدیدی به نام OperationWithArray ایجاد کنید.

2. یک کنترل Label، یک کنترل TextBox، یک کنترل CheckBox، سه کنترل RadioButton، سه کنترل ListBox و دو کنترل Button به فرم اضافه کنید:

3. متد random را به صورت زیر تعریف کنید:

```
void random(int[] a, int[] b, int n)
```

```
{
    Random r1 = new Random();
    Random r = new Random(r1.Next());
    for (int i = 0; i < n; i++) a[i] = r1.Next(100);
    for (int i = 0; i < n; i++) b[i] = r1.Next(100);
}
```

این متد، سه پارامتر آرایه **a**، **b** و **n** را گرفته **n** عدد تصادفی در آرایه **a** و آرایه **b** قرار می‌دهد.  
**4.** متد **find** را به صورت زیر تعریف کنید:

```
bool find(int a, int[] b, int n)
{
    for (int i = 0; i < n; i++)
        if (a == b[i]) return true;
    return false;
}
```

این متد، مقدار **a** و آرایه **b** که دارای **n** عنصر است را گرفته و مقدار **a** را در آرایه **b** جستجو می‌کند.  
**5.** متد **print** را به صورت زیر تعریف کنید:

```
void print(int[] a, int n, ListBox list1)
{
    list1.Items.Clear();
    for (int i = 0; i < n; i++) list1.Items.Add(a[i]);
}
```

این متد، آرایه **a** با **n** عنصر را گرفته در **listBox**ی به نام **list** اضافه می‌کند.  
**6.** متد **difference** را به صورت زیر تعریف کنید:

```
void difference(int[] a, int[] b, int[] c, int n, ref int k)
{
    k = -1;
    for (int i = 0; i < n; i++)
        if (find(a[i], b, n) == false) c[++k] = a[i];
}
```

این متد، دو آرایه به نام‌های **a** و **b** با طول **n** عنصر را گرفته در آرایه **c** قرار می‌دهد و تعداد عناصر آرایه **c** را که در **k** قرار می‌گیرد به تابع فراخوان برمی‌گرداند.

**7.** متد **share** را به صورت زیر تعریف کنید:

```
void share(int[] a, int[] b, int[] c, int n, ref int k)
{
    k = -1;
    for (int i = 0; i < n; i++)
        if (find(a[i], b, n) == true) c[++k] = a[i];
}
```

این متد، دو آرایه **a** و **b** با **n** عنصر را گرفته و اشتراک دو آرایه را در **c** قرار می‌دهد و تعداد عناصر آرایه **c** را از طریق **k** برمی‌گرداند.

**8.** متد **union** را به صورت زیر تعریف کنید:

```

void union(int[] a, int[] b, int[] c, int n, ref int k)
{
    k = -1;
    for (int i = 0; i < n; i++)
        if (find(a[i], c, k) == false) c[++k] = a[i];
    for (int i = 0; i < n; i++)
        if (find(b[i], a, n) == false) c[++k] = b[i];
}

```

این متد، دو آرایه a و b با n عنصر را گرفته، اجتماع آنها را در آرایه c قرار می‌دهد و تعداد عناصر آرایه c را از طریق پارامتر k برمی‌گرداند.

**9.** دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تایپ کنید:

```

private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt16(textBox1.Text);
    int[] num1 = new int[n];
    int[] num2 = new int[n];
    int[] num3 = new int[n+n];
    int k = 0;
    random(num1, num2, n);
    print(num1, n, listBox1);
    print(num2, n, listBox2);
    if (radioButton3.Checked == true && checkBox1.Checked ==
        false) difference(num1, num2, num3, n, ref k);
    if (radioButton3.Checked == true && checkBox1.Checked ==
        true) difference(num2, num1, num3, n, ref k);
    if (radioButton1.Checked == true) share(num1, num2, num3, n,
        ref k);
    if (radioButton2.Checked == true) union (num1, num2, num3, n,
        ref k);
    print(num3, k, listBox3);
}

```

دستوراول، تعداد عناصر (n) را می‌خواند. دستورات دوم و سوم، آرایه‌های به نام‌های num1 و num2 با n عنصر تعریف می‌کنند. دستور چهارم، آرایه‌ای به نام num3 با  $2 * n$  عنصر تعریف می‌نماید. دستور پنجم، k را تعریف کرده برابر صفر قرار می‌دهد. دستور ششم، اعداد تصادفی را تولید کرده در آرایه‌های num1 و num2 قرار دهد. دستور هفتم، اطلاعات آرایه num1 را در listBox1 و دستور هشتم، اطلاعات آرایه num2 را در listBox2 نمایش می‌دهد. دستور نهم، اگر radioButton3 انتخاب شده باشد و checkBox1 انتخاب نشده باشد، تفاضل آرایه num2 را از آرایه num1 حساب می‌کند. ولی اگر checkBox1 انتخاب شده باشد، تفاضل آرایه num1 را از آرایه num2 یک تعیین می‌نماید. دستور دهم، اگر radioButton1 انتخاب شده باشد، اشتراک دو آرایه را در num3 قرار می‌دهد. دستور یازدهم، اگر radioButton2 انتخاب شده باشد، اجتماع دو آرایه num1 و num2 را بدست آورده در آرایه num3 قرار می‌دهد و دستور دوازدهم، آرایه num3 را در listBox3 نمایش می‌دهد.

**10.** دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```

private void button2_Click(object sender, EventArgs e)
{
    Close();
}

```

```

    }

```

**11.** ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Form1\_Load را به صورت زیر تایپ کنید:

```

private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "Enter n";
    button1.Text = "Execute";
    button2.Text = "Exit";
    checkBox1.Text = "B - A";
    radioButton1.Text = "Share";
    radioButton2.Text = "Union";
    radioButton3.Text = "Difference";
}

```

**12.** پروژه را ذخیره و اجرا کنید. گزینه Difference را انتخاب کرده و جلوی Enter n عدد 20 را وارد کنید. سپس، دکمه Execute را کلیک کرده تا خروجی را به صورت زیر ببینید:

**مثال 10 - 3** برنامه‌ای که n را خوانده آرایه‌ای N\*N ایجاد می‌کند که، عناصر هر خانه برابر با شماره سطر \* شماره ستون می‌باشد. سپس، جمع سطرها و ستون‌ها را محاسبه می‌نماید و نمایش می‌دهد.

مراحل طراحی و اجرا

1. پروژه جدیدی به نام ArrayToDim ایجاد کنید.
2. دو کنترل Label، یک کنترل TextBox و دو کنترل Button به فرم اضافه کنید.

**3.** ناحیه خالی فرم را کلیک مضاعف کرده، دستورات رویداد Load آن را به صورت زیر تغییر دهید:

```

private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "Enter n";
    button1.Text = "Execute";
    button2.Text = "Close";
    label2.Text = "";
}

```

```
}
```

4. دکمه button1 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt16(textBox1.Text);
    int[,] a = new int[n + 1, n + 1];
    for (int i = 0; i < n; i++)
    {
        a[i, n] = 0 ;
        for (int j = 0; j < n; j++)
        { a[i, j] = (i+1) * (j+1) ;
          a[i, n] = a[i, n] + a[i, j];
        }
    }
    for (int i = 0; i < n; i++)
    {
        a[n, i] = 0 ;
        for (int j = 0; j < n; j++) a[n, i] = a[n, i] + a[j, i];
    }
    for (int i = 0; i <= n; i++)
    {
        for (int j = 0; j <= n; j++)
        {
            label2.Text = label2.Text + a[i, j].ToString() + " ";
        }
        label2.Text += "\n";
    }
}
```

دو دستور اول، n را خوانده و آرایه n+1 در n+1 را تعریف می‌کنند. حلقه اول، آرایه را مقداردهی کرده و مجموع عناصر سطرها را در ستون آخر قرار می‌دهد. حلقه دوم، مجموع عناصر ستون‌ها را حساب می‌نماید و در سطر آخر قرار می‌دهد و حلقه سوم، عناصر آرایه را بر روی Label2 نمایش می‌دهد.

5. دکمه button2 را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

6. پروژه را ذخیره و اجرا کنید. جلوی Enter n عدد 3 را وارد کرده، دکمه Execute را کلیک نمایید تا خروجی را به شکل زیر مشاهده کنید:

|   |    |    |    |
|---|----|----|----|
| 1 | 2  | 3  | 6  |
| 2 | 4  | 6  | 12 |
| 3 | 6  | 9  | 18 |
| 6 | 12 | 18 | 0  |

فروشگاهی را در نظر بگیرید که درصدی از فروش را به عنوان دستمزد به فروشنده می‌پردازد. میزان دستمزد با توجه به نوع کالا به صورت زیر محاسبه می‌شود:

کالاهای درجه 1: اگر میزان فروش کمتر یا مساوی 20000000 ریال باشد، 5 درصد به فروشنده پرداخت می‌شود، اگر میزان فروش بین 20000000 تا 200000000 ریال باشد، 7 درصد، وگرنه 10 درصد به فروشنده پرداخت می‌شود.

کالاهای درجه 2: اگر میزان فروش کمتر یا مساوی 20000000 ریال باشد، 4 درصد، وگرنه، 6 درصد به فروشنده پرداخت می‌گردد.

کالاهای درجه 3: 4 درصد کل فروش به فروشنده پرداخت خواهد شد.

کالاهای درجه 4: 6 درصد کل فروش به فروشنده پرداخت می‌گردد.

مراحل طراحی و اجرا

1. پروژه جدیدی به نام Sell ایجاد کنید.

2. دو کنترل Label به فرم اضافه کرده، خاصیت Text آن‌ها را به ترتیب "فروش" و "" تعیین کنید.

3. دو کنترل Button به فرم اضافه کرده، خاصیت Text آن‌ها را محاسبه و خروج تعیین نمایید.

4. چهار کنترل RadioButton به فرم اضافه کرده، خاصیت Text آن‌ها را به ترتیب درجه 1، درجه 2، درجه 3 و درجه 4 تعیین کنید.

5. دکمه محاسبه را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    double s = Convert.ToDouble(textBox1.Text);
    double pay = 0;
    if (radioButton1.Checked == true)
    {
```

```

if (s <= 20000000)
    pay = s * 5 / 100;
else if (s <= 200000000)
    pay = s * 7 / 100;
else
    pay = s * 10 / 100;
}
else if (radioButton2.Checked == true)
{
    if (s <= 20000000)
        pay = s * 4 / 100;
    else
        pay = s * 6 / 100;
}
else if (radioButton3.Checked == true)
{
    pay = s * 4 / 100;
}
else if (radioButton4.Checked == true)
{
    pay = s * 6 / 100;
}
}

```

این دستورات با توجه به انتخاب درجه کالا توسط کاربر ( از طریق radioButton ها)، دستمزد فروشنده را محاسبه کرده نمایش می دهند.

**6.** دکمه خروج را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```

private void button2_Click(object sender, EventArgs e)
{
    Close();
}

```

**7.** پروژه را ذخیره و اجرا کنید. نوع کالا را درجه 2 و مبلغ

فروش را 200000000 ریال وارد کرده و دکمه محاسبه را

کلیک نمایید تا خروجی روبه رو را ببینید.

برنامه‌ای که حقوق ناخالص را گرفته و براساس قوانین زیر، بیمه و مالیات را کسر کرده، حق تاهل به او تخصیص دهد:

❖ اگر نوع بیمه تامین اجتماعی باشد، 7 درصد حقوق ناخالص بیمه کسر گردد.  
❖ اگر نوع بیمه خدمات درمانی باشد، 9 درصد حقوق ناخالص بیمه باشد.  
❖ اگر نوع بیمه نیروی انتظامی باشد، 10 درصد حقوق ناخالص بیمه می‌باشد.  
❖ اگر کارمند جانباز باشد، از مالیات معاف است.  
❖ اگر کارمند منطقه نیمه توسعه یافته باشد، 5 درصد مزاد بر 4600000 ریال مالیات کسر می‌شود.  
❖ اگر کارمند عادی باشد، 10 درصد مزاد بر 4600000 ریال مالیات کسر می‌شود.  
❖ اگر کارمند متاهل باشد، 460000 ریال حق تاهل به او تعلق می‌گیرد.  
سپس با استفاده از فرمول زیر حقوق خالص را حساب کرده، نمایش می‌دهد:

مالیات - بیمه - حق تاهل + حقوق ناخالص = حقوق خالص

توضیح: در این برنامه دو کنترل GroupBox در نظر گرفته شده است و در هر یک از کنترل‌های GroupBox سه کنترل RadioButton اضافه شده‌اند. گروه اول، نوع بیمه و گروه دوم نوع کسر مالیات را تعیین می‌کند و یک CheckBox تعیین می‌کند که آیا کارمند متاهل است یا خیر.

#### مراحل طراحی و اجرا

1. پروژه جدیدی به نام Pay ایجاد کنید. خاصیت RightToLeft فرم را Yes تعیین کنید.
2. دو کنترل Label، دو کنترل Button و یک کنترل TextBox به فرم اضافه کرده، خاصیت Text کنترل‌های Label را به ترتیب حقوق ناخالص و خالی تعیین کنید. خاصیت Text کنترل‌های Button را به ترتیب محاسبه و خروج تعیین نمایید.
3. دو کنترل GroupBox به فرم اضافه کنید. خاصیت Text آن‌ها را به ترتیب نوع بیمه و نوع مالیات تعیین کنید.
4. یک کنترل CheckBox به فرم اضافه کرده، خاصیت Text آن را متاهل تعیین کنید.
5. سه کنترل RadioButton به کنترل groupBox1 (نوع بیمه) اضافه کرده، خاصیت Text آن‌ها را به ترتیب تامین اجتماعی، خدمات درمانی و نیروی انتظامی تعیین کنید.



6. سه کنترل RedioButton دیگر به کنترل نوع مالیات اضافه کرده، خاصیت Text آن‌ها را به ترتیب جانباز، نیمه توسعه یافته و عادی تعیین کنید.

7. دکمه محاسبه را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    double salary = Convert.ToDouble(textBox1.Text);
    double tax = 0, wife = 0, insuranc = 0;
    if (radioButton1.Checked == true)
        insuranc = salary * 7 / 100;
    else if (radioButton2.Checked == true)
        insuranc = salary * 9 / 100;
    else if (radioButton3.Checked == true)
        insuranc = salary * 10 / 100;
    if (radioButton4.Checked == true) tax = 0;
    else if (radioButton5.Checked == true)
        tax = (salary - 4600000) * 5 / 100;
    else if (radioButton6.Checked == true)
        tax = (salary - 4600000) * 10 / 100;
    if (checkBox1.Checked == true) wife = 4600000;
    double pay = salary + wife - insuranc - tax;
    label2.Text = "مالیات " + insuranc.ToString() + " " + "بیمه "
        + tax.ToString() + "\n";
    label2.Text += "پرداختی " + wife.ToString() + " " + "تاهل "
        + pay.ToString() ;
}
```

این دستورات ابتدا، حقوق ناخالص را خوانده و متغیرهای بیمه (insurance)، مالیات (tax) و حق تاهل (wife) را تعریف می‌کنند. سپس، با توجه به انتخاب کاربر و با استفاده از دو if متداخل، بیمه و مالیات را محاسبه می‌کنند. در پایان، اگر کارمند حق تاهل بگیر است، حق تاهل را برای او در نظر می‌گیرد و حقوق خالص را محاسبه کرده، بیمه، مالیات، حق تاهل و حقوق خالص را نمایش می‌دهد.

8. دکمه خروج را کلیک مضاعف کرده، دستورات رویداد Click آن را به صورت زیر تغییر دهید:

```
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

9. پروژه را ذخیره و اجرا کنید. جلوی حقوق ناخالص مقدار 10000000 را وارد کرده، گزینه‌های متاهل، تامین اجتماعی، نیمه توسعه یافته را انتخاب نمایید و دکمه محاسبه را کلیک کنید تا خروجی زیر را ببینید:

## مسائل آشنایی با زبان C#

### فصل 1

**مثال 1 - 1** برنامه‌ای که سه کنترل Button را به فرم اضافه کرده با کلیک دکمه button1، اندازه فرم حداکثر می‌گردد. با کلیک دکمه button2، اندازه فرم به حالت نرمال برمی‌گردد و با کلیک دکمه button3، اندازه فرم کمینه می‌شود. چنانچه مکان‌نمای ماوس از محدوده فرم خارج شود، رنگ زمینه فرم آبی می‌گردد و اگر مکان‌نما به محدوده فرم برگردد ( وارد شود)، رنگ زمینه فرم قرمز می‌شود.

**مثال 2 - 1** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ❖ عبارت "فرم اصلی" را در سمت راست عنوان فرم نمایش می‌دهد.
- ❖ با کلیک روی فرم، عنوان فرم به سمت چپ برده و با کلیک دیگر عنوان به سمت راست برود و این روند ادامه یابد.
- ❖ دارای دو کنترل TextBox و شش کنترل Button باشد.
- ❖ با کلیک button1، اطلاعات textBox2 به textBox1 اضافه می‌شود.
- ❖ با کلیک دکمه button2، اطلاعات textBox1 به textBox2 اضافه می‌گردد.
- ❖ با کلیک دکمه button3، اطلاعات انتخاب شده کنترل textBox2 حذف می‌گردد.
- ❖ با کلیک دکمه button4، رنگ زمینه فرم به سبز تغییر یابد.
- ❖ با کلیک دکمه button5، رنگ زمینه فرم به خاکستری تبدیل شود.
- ❖ با کلیک دکمه button6، یک تصویر در زمینه فرم قرار می‌گیرد.

**مثال 3 - 1** برنامه‌ای که اعمال زیر را انجام می‌دهد:

1. یک کنترل Label به فرم اضافه می‌کند.
2. اگر فرم کلیک گردد، عبارت Form Clicked بر روی label1 و عنوان فرم نوشته شود.
3. اگر مکان‌نما فرم را ترک کند، رنگ زمینه label سفید و رنگ زمینه فرم آبی شود.
4. اگر مکان‌نما وارد label شود، رنگ زمینه فرم قرمز و رنگ نوشته label سفید شود.
5. اگر بر روی فرم کلیدی فشرده شود، بر روی عنوان فرم و label عبارت Key Pressed نوشته شود.
6. اگر اندازه فرم تغییر یابد، در عنوان فرم و label1 عبارت Resizing با رنگ سبز نوشته شود.
7. وقتی مکان‌نما بر روی فرم حرکت می‌کند رنگ زمینه فرم سبز شده و در عنوان فرم و label1 عبارت Mouse Moved نوشته می‌شود.
8. اگر label کلیک گردید، برنامه خاتمه یابد.

**مثال 4 - 1** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ❖ یک کنترل TextBox و سه کنترل Button به فرم اضافه می‌نماید.
- ❖ با کلیک دکمه button1، محتویات TextBox را یک واحد به سمت راست انتقال می‌دهد.
- ❖ با کلیک دکمه button2، محتویات TextBox را یک واحد به سمت چپ انتقال می‌دهد.
- ❖ با کلیک دکمه button3، برنامه خاتمه می‌یابد.

**مثال 1 - 2** برنامه‌ای که حقوق ناخالص و نوع بیمه را دریافت می‌کند (حقوق ناخالص عددی و نوع بیمه به صورت RadioButton باشد) و با استفاده از فرمول زیر بیمه، مالیات و حقوق خالص را محاسبه می‌کند:

مالیات - بیمه - حقوق ناخالص = حقوق خالص

- ❖ اگر نوع بیمه 7٪ انتخاب شده باشد، 7 درصد حقوق ناخالص بیمه خواهد بود.
- ❖ اگر نوع بیمه 9٪ انتخاب شود، 9 درصد حقوق ناخالص بیمه خواهد بود.
- ❖ اگر نوع بیمه 10٪ انتخاب شود، 10 درصد حقوق ناخالص بیمه خواهد بود.
- ❖ اگر کنترل checkBox انتخاب نگردد و حقوق ناخالص بیش از 4500000 ریال باشد، مازاد بر این مبلغ 10٪ مالیات خواهد بود، وگرنه مالیات صفر منظور خواهد شد.
- ❖ در بخش حقوق از ورود داده‌های غیرعددی جلوگیری شود.

**مثال 2 - 2** فروشگاه‌ی را در نظر بگیرید که درصدی از فروش را به عنوان دستمزد به فروشنده می‌پردازد. میزان دستمزد با توجه به نوع کالا به صورت زیر محاسبه می‌شود:

- ❖ **کالاهای درجه 1:** اگر میزان فروش کمتر یا مساوی 20000000 ریال باشد، 5 درصد به فروشنده پرداخت می‌شود، اگر میزان فروش بین 20000000 تا 200000000 ریال باشد، 7 درصد، وگرنه 10 درصد به فروشنده پرداخت می‌شود.
- ❖ **کالاهای درجه 2:** اگر میزان فروش کمتر یا مساوی 20000000 ریال باشد، 4 درصد، وگرنه، 6 درصد به فروشنده پرداخت می‌گردد.

❖ **کالاهای درجه 3:** 4 درصد کل فروش به فروشنده پرداخت خواهد شد.

❖ **کالاهای درجه 4:** 6 درصد کل فروش به فروشنده پرداخت می‌گردد.

**مثال 3 - 2** برنامه‌ای که حقوق ناخالص را گرفته و براساس قوانین زیر، بیمه و مالیات را کسر کرده، حق تاهل به او تخصیص دهد:

❖ اگر نوع بیمه **تامین اجتماعی** باشد، 7 درصد حقوق ناخالص بیمه کسر گردد.

❖ اگر نوع بیمه **خدمات درمانی** باشد، 9 درصد حقوق ناخالص بیمه باشد.

❖ اگر نوع بیمه **نیروی انتظامی** باشد، 10 درصد حقوق ناخالص بیمه می‌باشد.

❖ اگر کارمند جانباز باشد، از مالیات معاف است.

❖ اگر کارمند منطقه نیمه توسعه یافته باشد، 5 درصد مازاد بر 4600000 ریال مالیات کسر می‌شود.

❖ اگر کارمند عادی باشد، 10 درصد مازاد بر 4600000 ریال مالیات کسر می‌شود.

❖ اگر کارمند متاهل باشد، 460000 ریال حق تاهل به او تعلق می‌گیرد.

سپس با استفاده از فرمول زیر حقوق خالص را حساب کرده، نمایش می‌دهد:

مالیات - بیمه - حق تاهل + حقوق ناخالص = حقوق خالص

**توضیح:** در این برنامه دو کنترل `GroupBox` در نظر گرفته شده است و در هر یک از کنترل‌های `GroupBox` سه کنترل `RadioButton` اضافه شده‌اند. گروه اول، نوع بیمه و گروه دوم نوع کسر مالیات را تعیین می‌کند و یک `CheckBox` تعیین می‌کند که آیا کارمند متاهل است یا خیر.

**مثال 4 - 2** برنامه‌ای که سن‌تان را به سال، ماه و روز گرفته به دقیقه تبدیل کرده و نمایش می‌دهد (هر سال 356 روز، هر ماه 30 روز و هر روز 24 ساعت است). در این برنامه کاربر می‌تواند از طریق کلیدهای ↑، ↓، ←، →، End، Home بین `TextBox`ها حرکت کند.

**مثال 5 - 2** برنامه‌ای که به دانش‌آموزان ضرب را آموزش می‌دهد. در این برنامه با استفاده از متد `next` کلاس `Random` یک سری اعداد تصادفی مثبت بین 0 تا 9 تولید می‌کند و سپس با استفاده از این اعداد تصادفی یک سوال به صورت زیر مطرح می‌نماید:

$5*4=?$

اکنون، دانش‌آموزان، جواب سوال را در یک `textBox` وارد می‌کند. اگر پاسخ صحیح باشد، پیغام "very Good!" را نمایش می‌دهد. وگرنه، پیغام "No.Please try again" را نمایش خواهد تا دانش‌آموزان جواب را دوباره وارد کند (برنامه متدی برای طرح سوال دارد).

**مثال 6 - 2** برنامه که اعتبار سنجی اطلاعات یک `TextBox` را انجام می‌دهد. یعنی کاربر می‌تواند تعیین کند چه اطلاعاتی را وارد کند. در این برنامه کاربر می‌تواند تعیین کند یکی از اطلاعات عددی، فقط کارکتر، پست الکترونیک و اعداد اعشاری وارد شود.

**مثال 7 - 2** برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل عبارت زیر را محاسبه نموده نمایش می‌دهد:

$$s = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \pm \frac{x^n}{n!}$$

**مثال 8 - 2** برنامه‌ای که  $n$  را از ورودی خوانده و مقدار سری زیر ( $\pi$ ) را حداکثر تا  $n$  جمله نمایش می‌دهد:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

**مثال 9 - 2** برنامه‌ای که مجموع سری زیر را نمایش می‌دهد ( $n$  را می‌خواند):

$$\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

**مثال 10 - 2** برنامه‌ای که کلیه اعداد سه رقمی را چاپ می‌کند که مجموع فاکتوریل ارقام آن اعداد برابر خودشان باشد (مانند فرمول زیر):

$$n3n2n1 = n3! + n2! + n1!$$

**مثال 11 - 2** برنامه‌ای که کلیه اعداد چهار رقمی را چاپ می‌کند که مجموع رقم اول به توان 1 و رقم چهارم به توان 4 برابر با مجموع رقم دوم به توان 2 و رقم سوم به توان 3 است. به عنوان مثال، عدد  $2141 = 2^4 + 1^1 = 1^3 + 4^2 = 17$  از جمله این اعداد می‌باشد.

**مثال 12 - 2** برنامه‌ای که عددی را خوانده، تشخیص می‌دهد این عدد جزء سری فیبوناچی است یا خیر.

**مثال 13 - 2** برنامه‌ای که کلیه اعداد چهار رقمی که توان دوم مجموع دو عدد حاصل از دو رقم سمت راست آن‌ها با دو رقم سمت چپ آن‌ها با خود آن عدد برابر است. به عنوان مثال،  $3025 = (30 + 25)^2$  می‌باشد.

**مثال 14 - 2** برنامه‌ای که دو عدد را خوانده، عدد اول را به توان عدد دوم می‌رساند (بدون استفاده از عملگر توان و ضرب). در این برنامه، از ورود داده‌های غیر عددی جلوگیری می‌کند.

**مثال 15 - 2** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ❖ دکمه‌ای که اطلاعات یک کنترل TextBox را به کنترل listBox1 انتقال می‌دهد.
- ❖ دکمه‌ای داشته باشد تا گزینه‌های انتخاب شده listBox1 را به listBox2 انتقال دهد.
- ❖ دکمه‌ای داشته باشد تا کلیه گزینه‌های listBox1 را به listBox2 انتقال دهد.
- ❖ دکمه‌ای داشته باشد تا اطلاعات listBox2 را حذف کند.

دکمه‌ای داشته باشد تا متن textBox1 را در مکانی که در textBox2 وارد شده است، اضافه کند

**مثال 16 – 2** عددی متقارن است که از هر طرف بخوانید یک عدد خوانده شود. به عنوان مثال، اعداد 4004، 21312، 12421 متقارن هستند. برنامه‌ای که یک عدد پنج رقمی را خوانده تشخیص می‌دهد، عدد متقارن است یا خیر. این برنامه دارای امکانات زیر می‌باشد:

- ❖ برنامه فقط ارقام را می‌پذیرد.
- ❖ اگر کاربر عدد 5 رقمی وارد نکند، پیغام مناسبی نمایش می‌دهد.
- ❖ **مثال 17 – 2** فرض کنید شرکت مخابرات یکی از استان‌ها می‌خواهد اطلاعات خودش را از طریق خط تلفن منتقل نماید. تمام اطلاعات انتقالی 8 رقمی (فقط ارقام) می‌باشند. شرکت مخابرات می‌خواهد برای ارسال آن را رمزگذاری نماید، به طوری که هر رقم را به توان عددی که در textBox2 وارد شده است، نماید و باقی‌مانده آن را بر عدد وارد شده در textBox3 محاسبه می‌کند و جایگزین آن رقم می‌کند و اطلاعات رمزگذاری شده را مغلوب کرده نمایش دهد.

❖ **مثال 18 – 2** برنامه‌ای که تصویری را بر روی pictureBox1 قرار می‌دهد و سپس می‌توانید آن را با کشیدن و رها کردن به pictureBox2 انتقال دهید.

مثال 19 – 2 برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ❖ دکمه‌های کمینه، بیشینه و بستن فرم را غیر فعال می‌کند.
- ❖ تصویر خاصی را به کنترل TextBox انتقال می‌دهد.
- ❖ خاصیت Text تعدادی کنترل را با متدی پاک می‌کند.
- ❖ خاصیت Text کلیه کنترل‌های روی فرم را به صورت بازگشتی حذف می‌کند.
- ❖ فرم را به مرور محو کرده، از برنامه خارج می‌شود.

**20 – 2** برنامه‌ای که n را خوانده جذر n را تا دقت  $10^{-10}$  محاسبه کرده نمایش می‌دهد.

**مثال 21 – 2** برنامه‌ای که n را خوانده حاصل عبارت  $\sqrt{n + \sqrt{n + \sqrt{n + \dots}}}$  را نمایش می‌دهد.

**مثال 23 – 2** برنامه‌ای که x و n را خوانده و حاصل سری زیر را نمایش می‌دهد.

$$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{2*4} + \frac{x^5}{3*5} + \dots \pm \frac{x^n}{1*3*5*...*n}$$

## مسائل آرایه‌ها و رشته‌ها

### فصل 3

**مثال 1-3** برنامه‌ای که عدد صحیح  $n$  را خوانده تمام اعداد اول قبل از آن را در یک listBox نمایش می‌دهد ( عددی اول است که بر هیچ عدد اول کوچکتر از خودش، بخش پذیر نباشد ).

**مثال 2-3** برنامه‌ای که 100 عدد تصادفی بین 0 تا 99 تولید می‌کند. سپس، تعیین می‌کند هر عدد چند بار تکرار شده است.

**مثال 3-3** برنامه‌ای، که حداکثر 100 عدد تصادفی بین 0 تا 99 تولید کرده در آرایه‌ای قرار می‌دهد و سپس عددی که بیشترین تکرار را دارد، به همراه تعداد تکرار آن نمایش می‌دهد. اعداد را ابتدا مرتب می‌نماید.

**مثال 4-3** برنامه‌ای که 100 عدد تصادفی بین 0 تا 10 ایجاد کرده آنها را در آرایه‌ای قرار می‌دهد سپس، کلیه عناصر آرایه را بر عنصر وسط تقسیم می‌کند. اگر عنصر وسط صفر باشد، بر عنصر بعد از عنصر وسط تقسیم می‌نماید. اگر این عنصر صفر باشد، بر عنصر قبل از عنصر وسط تقسیم می‌کند. اگر این عنصر صفر باشد، برای پیدا کردن عنصری غیر از صفر و انجام تقسیم، به روند قبلی ادامه می‌دهد. اگر همه عناصر آرایه صفر باشند، پیام مناسبی صادر می‌کند. پس از قرار دادن عناصر در آرایه، اندیس وسط را محاسبه کرده، عنصر وسط را پیدا می‌نماید. اگر صفر نبود، عمل تقسیم را انجام می‌دهد. این شمارنده به اندیس وسط اضافه می‌شود تا اگر مثبت بود، عنصر بعد از عنصر وسط و اگر منفی بود، عنصر قبل از عنصر وسط مورد بررسی قرار گیرد تا عمل تقسیم انجام شود. اگر عنصر غیر صفری در آرایه پیدا شود، action فراخوانی می‌شود تا تمام عناصر آرایه را بر آن عنصر تقسیم کند. اگر همه عناصر صفر باشند، برنامه پیام مناسبی صادر می‌کند.



**مثال 5 - 3** برنامه‌ای که  $n$  عدد تصادفی تولید کرده در آرایه‌ای قرار می‌دهد. سپس، عناصر آرایه را به اندازه  $m$  عنصر به سمت چپ یا راست شیفت می‌دهد. در این برنامه، برای تعیین نوع شیفت از کنترل `RadioButton` استفاده می‌شود.

**مثال 6 - 3** یک عدد صحیح اول است که بر خودش و یک بخش‌پذیر باشد. غربال اراتوستن روشی برای تعیین عدد اول است. در این روش، آرایه‌ای در نظر می‌گیرد و اعمال زیر را بر روی آن انجام می‌دهد:

1. مقدار اولیه همه عناصر آرایه را برابر 1 قرار می‌دهد.  
2. با شروع از اندیس 2 (اندیس 1 حتما اول است)، هر بار که عنصری با مقدار 1 یافت می‌شود، بقیه عناصری که اندیس‌شان مضربی از 2 باشد (اندیس‌های 4، 6، 8، 10 و 000) را برابر صفر قرار می‌دهد. برای اندیس‌های 3، همه عناصر بعد از 3 که اندیس‌شان مضربی از 3 است (اندیس‌های 6، 9، 12، 15، 000) صفر قرار می‌گیرد، و به همین ترتیب، این روند ادامه می‌یابد.

3. در پایان عناصری که هنوز یک هستند، اندیس‌های‌شان اولند.  
برنامه‌ای که با استفاده از یک آرایه 1000 تایی، اعداد اول 1 تا 999 را بدست آورده و نمایش می‌دهد.

**مثال 7 - 3** برنامه‌ای که فاکتوریل اعداد بزرگ را محاسبه می‌کند. به عنوان مثال، فاکتوریل 30 را محاسبه می‌کند که حاصل آن در یک نوع `double` جاها نمی‌شود.

**مثال 8 - 3** برنامه‌ای که عددی را خوانده اعداد متحابه را از 1 تا عدد خوانده شده نمایش می‌دهد. دو عدد را متحابه گویند که مجموع مقسوم علیه‌های هر یک به جزء خود عدد با عدد دوم برابر باشد. به عنوان مثال، مقسوم علیه‌های 220 که از 220 کوچکتر هستند، عبارت‌اند از: 5، 110، 44، 22، 10، 5، 4، 2، 1، که مجموع آن‌ها عدد 248 می‌شود و مقسوم علیه‌های 248 نیز کوچکتر از 248 باشند، برابراند با 142، 71، 4، 2، 1 که مجموع آن‌ها 220 است. پس این دو عدد متحابه هستند

**مثال 9 - 3** برنامه‌ای که دو آرایه  $n$  عنصری ( $n$  را از ورودی می‌خواند) تعریف کرده توسط تابع تصادفی اعداد بین 0 تا 99 تولید کرده در این دو آرایه قرار می‌دهد. سپس، اعمال زیر را انجام می‌دهد:

- ❖ متدی عناصر مشترک دو آرایه را پیدا می‌کند.
- ❖ متدی اجتماع دو آرایه را پیدا می‌کند.
- ❖ متدی تفاضل دو آرایه را برمی‌گرداند (توسط یک `CheckBox` تعیین می‌گردد که کدام آرایه از آرایه دوم کم گردد).
- ❖ برای این که تعیین شود کدام یک از اعمال انجام گردد، از سه `RadioButton` استفاده می‌شود.

❖ **مثال 10 - 3** برنامه‌ای که  $n$  را خوانده آرایه‌ای  $N*N$  ایجاد می‌کند که، عناصر هر خانه برابر با شماره سطر \* شماره ستون می‌باشد. سپس، جمع سطرها و ستون‌ها را محاسبه می‌نماید و نمایش می‌دهد.

**مثال 11 - 3** برنامه‌ای که یک تاس را 200 بار پرتاب می‌کند تعیین می‌کند اعداد 1، 2، 3، 4، 5 و 6 چند بار تکرار شده‌اند (برای تولید پرتاب تاس از یک تابع تصادفی که عددی بین 1 تا 6 تولید می‌کند، استفاده می‌نماید).

**مثال 12 - 3** برنامه‌ای که  $n$  را خوانده، تمام جایگشت‌های عدد 1 تا  $n$  را نمایش می‌دهد.

**مثال 13 - 3** یک شرکت هواپیمایی قصد دارد، سیستم ذخیره و فروش بلیط را کامپیوتری کند. برنامه سیستم رزرو بلیط مراحل زیر را انجام می‌دهد (تعداد صندلی 40 تا می‌باشد که شماره 1 تا 20 برای سیگاری‌ها و از 21 تا 40 برای غیرسیگاری‌ها در نظر گرفته می‌شود. در ضمن حداکثر 10 شماره پرواز از 1 تا 10 داریم):  
الف. آرایه دو بعدی با 40 سطر و 10 ستون در نظر گرفته شده که ستون‌های 0 تا 9 شماره پروازهای 1 تا 10 را تعیین می‌کنند.

ب. شماره مسافر، نوع سیگاری و غیرسیگاری (یک ChekBox) و شماره پرواز از کاربر دریافت می‌شود. سپس، بلیط را رزرو می‌نماید.

ج. اگر نوع مسافر سیگاری باشد، با توجه به شماره پرواز، به آن شماره پرواز در بخش سیگاری صندلی تخصیص می‌دهد. در صورت پر بودن بخش سیگاری از مشتری سوال کند که آیا در بخش غیرسیگاری‌ها پذیرفته شود یا خیر و برعکس.

د. شماره مسافر به شماره صندلی در آرایه تخصیص می‌یابد.

**مثال 14 - 3** ماتریس جادویی، ماتریس  $N*N$  است که عناصر آن اعداد صحیح بین 1 تا  $N^2$  است ( $N$ ، عددی صحیح و فرد است)، در این ماتریس حاصل جمع هر سطر، هر ستون و همچنین عناصر قطر همگی برابرند. برای ایجاد چنین ماتریس به روش زیر عمل می‌شود:

❖ عدد یک در ستون میانی اولین سطر قرار می‌گیرد.

❖ به طور مورب به سمت چپ و بالا حرکت می‌کند.

❖ اگر این خانه پر باشد، یک خانه به طرف پایین حرکت می‌کند و عدد بعدی را در آن مکان (خانه) قرار می‌دهد.

❖ اگر در حرکت به صورت مورب، از محدوده ماتریس خارج شد، باید به آخرین عنصر در آن سطر یا ستون برگردد.

❖ اگر از محدود سطر و ستون (هر دو) خارج شد، یک خانه به سمت پایین حرکت می‌کند.

برنامه‌ای که ماتریس جادویی را تولید کرده نمایش می‌دهد. نمونه‌ای از این ماتریس  $3*3$  در زیر آمده است:

|   |   |   |
|---|---|---|
| 5 | 1 | 9 |
| 8 | 4 | 3 |
| 2 | 7 | 6 |

**مثال 15 – 3** برنامه‌ای که جملات را به زبان انگلیسی گرفته به زبان خوکی تبدیل کرده نمایش می‌دهد. برای تبدیل لغات به زبان خوکی باید اولین کارکتر کلمه را به آخر آن انتقال داده به آن "ay" اضافه کنید. به عنوان مثال، کلمه **are** در زبان انگلیسی به کلمه **reaay** در زبان خوکی تبدیل خواهد شد (بین کلمات در رشته ورودی فاصله می‌باشد).

**مثال 16 – 3** رشته قرینه<sup>1</sup>، رشته‌ای است که از هر طرف بخوانید، یک جمله خوانده شود. به عنوان مثال، رشته "able was I ere I saw elba" قرینه است. برنامه، متدی به نام **isPalinRome** دارد که یک رشته را گرفته تشخیص می‌دهد قرینه است یا خیر. برنامه‌ای که یک رشته گرفته با استفاده از این متد اگر رشته قرینه باشد، **Yes**، وگرنه، **No** را نمایش می‌دهد.

**مثال 17 – 3** برنامه‌ای که عدد یا تاریخ شمسی را خوانده معادل فارسی آن‌ها را نمایش می‌دهد.

**مثال 18 – 3** برنامه‌ای که  $n$  را خوانده آرایه‌ای  $n*n$  تعریف می‌نماید. سپس این آرایه را با اعداد تصادفی 0 تا 100 پر می‌کند. در پایان آرایه را 90 درجه به سمت چپ چرخش می‌دهد.

**مثال 19 – 3** برنامه‌ای که بازی دوز را شبیه‌سازی می‌کند. در این بازی یک ماتریس  $3 \times 3$  داریم که دو بازیکن 1 و 2 با هم بازی می‌کنند. هر یک از این بازیکنان سه مهره دارند (هر سه مهره شماره همان بازیکن را دارد). بازیکنی برنده است که بتواند سریعتر مهره‌های خود را به صورت سطری یا ستونی و یا قطری ردیف کند. به عنوان مثال، در ماتریس زیر بازیکن شماره 2 برنده است:

|   |   |   |
|---|---|---|
|   |   | 2 |
|   | 2 | 1 |
| 2 | 1 | 1 |

---

<sup>1</sup> -palindrome

مثال 1 – 4 کلاسی برای اعداد کسری که امکانات زیر را دارد:

1. سازنده‌ای که از مخرج کسر صفر جلوگیری کند.
2. متدی برای جمع دو کسر دارد.
3. متدی برای تفریق دو کسر دارد.
4. متدی برای ضرب دو کسر دارد.
5. عملگر تقسیم را برای دو کسر مجددا تعریف می‌کند.
6. متدی برای ساده کردن دو کسر دارد.
7. تابع ToString را مجددا پیاده‌سازی می‌نماید تا یک عدد کسری را به صورت رشته برگرداند.

مثال 2 – 4 کلاسی به نام Complex که محاسبات بر روی اعداد موهومی ( $x + iy$ ) را انجام می‌دهد. این کلاس دارای امکانات زیر است:

- ❖ سازنده‌های دارد که نمونه‌های مورد نیاز را ایجاد می‌نماید.
- ❖ متدی برای ضرب دو عدد موهومی دارد.
- ❖ عملگر + را برای محاسبه مجموع دو عدد موهومی مجددا پیاده‌سازی می‌کند.
- ❖ عملگر - را برای تفریق دو عدد موهومی مجددا پیاده‌سازی می‌کند.
- ❖ متد ToString() را برای نمایش اعداد موهومی مجددا پیاده‌سازی می‌کند.
- ❖ متد Equals را برای مقایسه دو عدد موهومی پیاده‌سازی می‌کند.

مثال 3 – 4 برنامه‌ای که بزرگترین مقسوم علیه دو عدد را حساب می‌کند. این برنامه دارای تابع بازگشتی به نام GCD می‌باشد که با توجه به فرمول زیر بزرگترین مقسوم علیه دو عدد  $x$  و  $y$  را محاسبه می‌کند:

$$\text{GCD}(x, y) = \begin{cases} x & \text{اگر } y = 0 \\ \text{GCD}(y, x \% y) & \text{در غیر این صورت} \end{cases}$$

**مثال 4 - 4** کلاسی به نام Date بنویسید که دارای ویژگی‌های زیر باشد:

- ❖ سال، ماه و روز را بپذیرد.
- ❖ سازنده‌ای بنویسید که اگر ماه بین 1 تا 6 است، کاربر نتواند روز را بیشتر از 31 وارد نماید. ولی، اگر ماه بین 7 تا 11 است. کاربر نتواند روز را بیشتر از 30 وارد نماید و اگر ماه 12 بود، کاربر نتواند روز را بیشتر از 29 وارد کند. در این سازنده از ورود سال، ماه و روز منفی جلوگیری می‌کند.
- ❖ متد Equals را مجدداً پیاده‌سازی کنید تا دو تاریخ را با یکدیگر مقایسه کند.
- ❖ متدی به نام nextDay بنویسید که تاریخ را یک روز اضافه کند.
- ❖ متدی بنویسید که حاصل جمع دو تاریخ را برگرداند.
- ❖ عملگر - مجدداً تعریف کنید تا دو تاریخ را از هم تفریق کند.
- ❖ متد ToString() را مجدداً پیاده‌سازی کنید تا تاریخ را به فرمت روز/ماه/سال تبدیل کرده برگرداند (سال چهارم رقم، ماه دوم رقم و روز دوم رقم).

**مثال 5 - 4** در بازی Taxman یک کاربر در مقابل کامپیوتر بازی می‌کند. در این بازی، کاربر یک عدد بین 1 تا 100 را انتخاب می‌کند. اگر این عدد در دسترس باشد (یعنی، قبلاً استفاده نشده باشد و مقسوم علیه‌های عدد انتخاب شده نباشد)، کاربر به اندازه عدد انتخاب شده و کامپیوتر به اندازه مقسوم علیه‌های عدد وارد شده امتیاز می‌گیرد. کاربر همواره باید عددی را انتخاب کرده، که دارای مقسوم علیه باشد. پس از انتخاب عدد آن عدد و مقسوم علیه‌های آن از دسترس خارج می‌شوند. در پایان به اندازه مجموع تمام اعداد در خواست نشده توسط کاربر، کامپیوتر امتیاز می‌گیرد.

فرض کنید، بازی با اعداد 1 تا 6 انجام می‌شود. اگر کاربر عدد 6 را انتخاب کند، مجموع مقسوم علیه‌های آن یعنی،  $1+2+3$  به امتیاز کامپیوتر اضافه می‌گردد. بقیه اعداد یعنی، 4 و 5 هستند که مقسوم علیه‌های 4، یعنی 1 و 2 از مجموع حذف شده‌اند و 5 جزء خودش و 1 مقسوم علیه دیگر ندارد، در نتیجه کامپیوتر 15 و 6 برنده می‌شود. کاربر برای برنده شدن باید ابتدا عدد 5 را انتخاب کند و به دنبال آن عدد 4 و سپس 6 را برگزیند. در این صورت کاربر 15 بر 7 برنده خواهد شد. در این برنامه کلاس به نام Taxman طراحی کنید که دارای اعضای زیر باشد:

- ❖ امتیاز کاربر و کامپیوتر
- ❖ عدد ورودی
- ❖ تابع عضوی که مجموع مقسوم علیه‌های در دسترس را برمی‌گرداند.
- ❖ تابع عضوی که امتیاز کاربر و کامپیوتر را محاسبه می‌کند.
- ❖ تابع عضوی که مجموع اعداد باقی مانده (انتخاب شده) را به امتیاز کامپیوتر اضافه می‌کند.

❖ تابع سازنده‌ای که امتیاز کاربر و کامپیوتر را برابر صفر قرار داده تمام اعداد 1 تا 100 را در دسترس قرار می‌دهد (برای در دسترس قرار دادن اعداد از آرایه استفاده شده است. اگر اعداد در دسترس نباشند، مقدار مربوط به آن عدد در اندیس آرایه برابر صفر می‌گردد).

#### مثال 6 - 4 کلاسی با ویژگی‌های زیر بنویسید:

- ❖ هر شی از این کلاس می‌تواند اعداد صحیح بین 0 تا 100 را بپذیرد.
- ❖ این مجموعه به صورت آرایه‌ای از 0 و یک نمایش داده می‌شود.
- ❖ عدد صحیحی  $i$  را خوانده اگر این عدد صحیح در بازه 0 تا 100 باشد،  $a[i]$  برابر یک خواهد شد.
- ❖ سازنده‌ای که مقادیر آرایه را برابر صفر می‌کند.
- ❖ متدی که عنصر  $k$ ام را در آرایه درج می‌کند. یعنی،  $a[k]$  را برابر 1 قرار می‌دهد.
- ❖ متدی که عنصر  $k$ ام از آرایه را حذف می‌کند. یعنی،  $a[k]$  را برابر صفر قرار می‌دهد.
- ❖ متدی که اجتماع دو مجموعه را محاسبه می‌کند.
- ❖ متدی که اشتراک دو مجموعه را محاسبه می‌کند.
- ❖ متدی که دو مجموعه را مقایسه کند. یعنی، متد `Equals()` را مجدداً پیاده‌سازی می‌کند.
- ❖ متد `ToString()` را مجدداً پیاده‌سازی می‌کند که مقادیر مجموعه را با 4 کارکتر فاصله از یک دیگر قرار داده برمی‌گرداند.

#### مثال 7 - 4 کلاسی برای اعداد دهدهی با طول بلند بنویسید که دارای امکانات زیر باشد:

1. کلاس اعداد 100 رقمی را دریافت کند.
2. کلاس دارای سازنده‌ای باشد که تمام ارقام را صفر کند.
3. کلاس دارای متدی باشد که دو عدد 100 رقمی را جمع نماید.
4. کلاس دارای متدی است که دو عدد بزرگ را از هم تفریق می‌کند.
5. از ورود اطلاعات غیر رقم جلوگیری کند.

مثال 8 - 4 برنامه‌ای که پنج کلاس `Shape` (شکل)، `Square` (مربع)، `Rectangle` (مستطیل)، `Diamond` (لوزی) و `Rhomboid` (متوازی الاضلاع) را پیاده‌سازی می‌کند. این کلاس‌ها دارای ویژگی‌های زیر می‌باشند:

- ☒ کلاس `Shape` کلاسی از نوع `abstract` می‌باشد که پایه کلاس `Square` است.
  - ☒ کلاس `Square`، پایه کلاس‌های `Rectangle` و کلاس `Diamond` می‌باشد.
  - ☒ کلاس `Rectangle` پایه کلاس `Rhomboid` است.
- هدف این برنامه پیاده‌سازی کلاس‌های `abstract`، پیاده‌سازی مفهوم مشتق، ایجاد سازنده‌های کلاس مشتق و پایه، پیاده‌سازی مجدد عملگر + و پیاده‌سازی متدهای `override` می‌باشد.

#### مثال 9 - 4 کلاسی برای کارمندان طراحی کنید که دارای امکانات زیر باشد:

☒ شماره کارمندی

☒ نام

☒ نام خانوادگی

در این سیستم سه نوع کارمند داریم که عبارت‌اند از:

☒ کارمندان رییس حقوق را به صورت ماهیانه دریافت می‌کنند (مبلغ حقوق آن‌ها ثابت است).

☒ کارمندان دیگر، براساس مبلغ فروش حقوق دریافت می‌نمایند. یعنی، برای آن‌ها نرخ دریافت حقوق (درصد) و مبلغ فروش را دریافت می‌کنیم تا حقوق آن‌ها با فرمول  $100 / \text{مبلغ فروش} * \text{نرخ دریافت حقوق}$  محاسبه شود.

☒ کارمندان نوع سوم، براساس ساعت کارکرد و نرخ هر ساعت کارکرد حقوق می‌گیرند. حقوق این کارمندان برابر با  $\text{نرخ هر ساعت} * \text{ساعت کارکرد}$  می‌باشد.

کلاس‌های را طراحی کنید که اطلاعات این کارمندان را نگهداری کند. در این کلاس به سوالات زیر توجه کنید:

☒ کلاس‌های پایه، abstract و مشتق را تعیین کنید.

☒ فیلدها و خواص هر کلاس را تعیین کنید.

☒ متد ToString را برای کل کلاس‌ها پیاده‌سازی کنید به طوری که نام، نام خانوادگی و حقوق کارمند را نمایش دهد.

☒ سازنده‌های هر کلاس را پیاده‌سازی کنید.

☒ دو نمونه از کلاس کارمند ساعتی ایجاد

کرده با عملگر + حقوق آن‌ها را جمع زده

برگردانید.

**مثال 10 – 4** برنامه‌ای که اطلاعات TextBox را به صورت عددی دریافت می‌کند و هر سه رقم را با کاما از هم جدا می‌کند.

**مثال 1 - 5** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ❖ کاربر می‌تواند کامپیوتر را Shutdown، log off یا Restart نماید.
- ❖ کاربر می‌تواند لیست پردازش‌ها (Process) را ببیند و یکی از آن‌ها را حذف کند.
- ❖ کاربر می‌تواند فایل اجرایی خاصی را انتخاب کرده، اجرا نماید.

**مثال 2 - 5** برنامه‌ای که تبدیل تاریخ میلادی به شمسی و برعکس را انجام می‌دهد.

**مثال 3 - 5** برنامه‌ای که امکان تغییر تصویر دسک‌تاپ کامپیوتر را فراهم می‌کند.

**مثال 4 - 5** برنامه‌ای که ارتباط با پورت سری کامپیوتر را برقرار می‌کند و اطلاعاتی را برای پورت‌ها می‌فرستد.

**مثال 5 - 5** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ❖ تصویری را باز می‌کند. تصویر را لایه گذاری کرده یا شفافیت رنگ آن را تغییر می‌دهد. تصویر را ذخیره می‌نماید و تصویر اولیه را برمی‌گرداند.
- ❖ از صفحه کامپیوتر عکس می‌گیرد.
- ❖ رنگ‌ها را ترکیب می‌کند و رنگ‌های ترکیب شده را نمایش می‌دهد.

**مثال 6 - 5** برنامه‌ای که عمل زیر را انجام می‌دهد.

- ❖ دکمه‌ای درایو CD را باز می‌کند.
- ❖ دکمه‌ای دیگر درایو CD را می‌بندد.

**مثال 7 - 5** برنامه‌ای که فایلی را فشرده کرده یا از حالت فشرده خارج می‌کند.



**مثال 9 - 5** برنامه‌ای که درایوهای موجود در سیستم را در یک comboBox نمایش می‌دهد. کاربر می‌تواند درایو مورد نظر را انتخاب کرده، اطلاعات آن را ببیند.

**مثال 10 - 5** برنامه‌ای که آیکن خاصی را به طور تصادفی بر روی نقاط مختلف صفحه نمایش جابه‌جا می‌کند.

**مثال 11 - 5** برنامه‌ای که یک دایرکتوری را به صورت بازگشتی در دایرکتوری دیگر کپی می‌کند.

**مثال 12 - 5** برنامه که فرم خوش‌آمدگویی طراحی می‌کند و آن را برای چند لحظه نمایش می‌دهد.

**مثال 13-5** برنامه‌ای که اعمال زیر را انجام می‌دهد:

1. کنترل را از یک فرم می‌کشد و بر روی فرم دیگر رها می‌کند. در فرمی که کنترل بر روی آن رها می‌گردد، مشخصات کنترل رها شده (Draged) را نمایش می‌دهد.
2. مشخصات کنترل که مکان‌نمای ماوس بر روی آن رفته است را در نوار وضعیت (statustBar) فرم اصلی نمایش می‌دهد.

**مثال 14 - 5** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ☒ دکمه‌های با حالت مختلف رسم می‌کند.
  - ☒ چک باکس‌های (checkBox) مختلف ترسیم می‌نماید.
  - ☒ فرمی به شکل دایره رسم می‌کند.
  - ☒ رنگ زمینه کلیه فرم‌های باز را تغییر می‌دهد.
- ☒ **مثال 15 - 5** برنامه‌ای که چگونگی ایجاد و ذخیره کلیدهای میانبر (ShortCut) را در C# نشان می‌دهد.

**مثال 16 - 5** برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ☒ ایستگاه کاری (کامپیوتر فعلی) را قفل می‌کند.
- ☒ سطل بازیافت را خالی می‌نماید.
- ☒ زمان اجرای بخشی از برنامه را محاسبه می‌کند.
- ☒ برنامه را مجدداً اجرا می‌کند.
- ☒ بر روی دسک‌تاپ ویندوز متنی را می‌نویسد.
- ☒ شفافیت کنترل را تغییر می‌دهد.

**مثال 17 - 5** برنامه‌ای که متن با جلوه ویژه نمایش می‌دهد.

**مثال 18 - 5** برنامه‌ای که اعمال زیر بر روی تصویر خاصی را انجام می‌دهد:

- ☒ تصویری را بار می‌کند.
- ☒ تصویر را چرخش می‌دهد.
- ☒ وسط یک کنترل می‌نویسد.
- ☒ اندازه تصویر را تغییر می‌دهد.
- ☒ تصویر را کپی می‌نماید.

## پروژه‌های حل شده در سایت

1. پروژه‌ای که اعمال زیر را انجام می‌دهد:
  - ✖ پردازش‌های (Proccess‌های) در حالت اجرا بر روی کامپیوترتان را نمایش می‌دهد.
  - ✖ می‌توانید اجرای پردازش‌های در حال اجرا را خاتمه دهید.
  - ✖ می‌توانید خواص پردازش‌های در حال اجرا را ببینید.
  - ✖ می‌توانید اعمالی از قبیل خاموش کردن، راه‌اندازی مجدد و انتقال به حالت StandBy را انجام دهید.
  - ✖ می‌توانید کاربر فعلی را Logoff کرده یا کاربر فعلی را تغییر دهید.
  - ✖ می‌توانید یک فایل اجرا کنید.
2. پروژه‌ای که اعمال زیر را انجام می‌دهد:
  - ✖ پورت‌های Com موجود در کامپیوتر را شناسایی می‌کند.
  - ✖ می‌توانید بر روی پورت‌های موجود بنویسید یا از آن بخوانید.
  - ✖ مودم نصف شده در کامپیوترتان را شناسایی می‌کند.
  - ✖ می‌توانید فرمان‌های مختلف را برای مودم نصب شده بر روی کامپیوترتان اجرا کنید.
  - ✖ می‌توانید پورت‌های Com را غیر فعال کنید.
3. پروژه‌ای که تویی را بر روی فرم نمایش می‌دهد. اگر کاربر بر روی توپ کلیک کند، توپ به طور تصادفی در ناحیه‌های مختلف فرم حرکت می‌کند.
4. پروژه‌ای که پردازش تصاویر را انجام می‌دهد. این پروژه دارای امکانات زیر می‌باشد:
  - ✖ کاربر می‌تواند تصویر را انتخاب کرده و بر روی فرم انتقال دهد.
  - ✖ کاربر می‌تواند تصویر را تحلیل (آنالیز) کند. یعنی، بایت‌های تشکیل دهنده تصویر را ببیند:
  - ✖ کاربر می‌تواند تصاویر مختلف را با هم از نظر اندازه و محتوی مقایسه کند.

5. برنامه‌ای که اعمال زیر را انجام می‌دهد:

- ☒ اطلاعات کنتور قدیم را از یک فایل Text گرفته به یک فیلد DataGridView منتقل می‌کند.
- ☒ اطلاعات کنتور جدید را از یک فایل Text خوانده به فیلد دیگر DataGridView انتقال می‌دهد.

☒ کنتور جدیدی را از قدیم کم کرده محاسباتی بر روی آن‌ها انجام می‌دهد و به فیلد دیگر DataGridView منتقل می‌کند.

- ☒ اطلاعات را در یک فایل اکسس ذخیره می‌نماید.
- ☒ هدف از این پروژه آشنایی با فایل‌ها (Stream)، انتقال فایل Text به بانک اطلاعاتی، کارکردن با فضای نام OleDb و چاپ اطلاعات می‌باشد.

6. برنامه‌ای بانک اطلاعاتی برای مطب پزشک ایجاد می‌کند. این برنامه دارای امکانات زیر می‌باشد:

1. اطلاعات بیمار از قبیل کد کامپیوتری، نام بیمار، نام خانوادگی، نوع بیمه، کد دفترچه درمانی، آدرس، توضیحات، تلفن و شماره موبایل را گرفته در بانک اطلاعات ذخیره می‌نماید.
2. اطلاعات ویزیت از قبیل کد ویزیت، تاریخ ویزیت، شرح حال و غیره را دریافت کرده در بانک اطلاعات ذخیره می‌نماید.
3. اطلاعات کامل از قبیل کد بیمار، مشخصات بیمار، و اطلاعات ویزیت آن را دریافت کرده نمایش می‌دهد.

4. در این برنامه امکان Login و ایجاد کاربر جدید وجود دارد.

5. این برنامه امکانات تهیه گزارش را دارد. البته این برنامه کامل نشده است و هدف آن آشنایی دانشجویان عزیز با برنامه‌های بانک اطلاعات می‌باشد.

7. پروژه‌ای که یک بانک اطلاعات پویا (Dynamic Database) ایجاد می‌کند. این پروژه دارای امکانات زیر می‌باشد:

- ☒ تشخیص سرویس دهنده‌های SQL Server نصب شده بر روی کامپیوترتان یا موجود در شبکه.
- ☒ دریافت مشخصات اتصال و اتصال به بانک اطلاعات به روش احراز هویت ویندوزی یا SQL Server ای. اطلاعات احراز هویت و مشخصات اتصال را از کاربر دریافت می‌کند.
- ☒ ایجاد بانک اطلاعات جدید به صورت پویا، مشخصات بانک اطلاعاتی از قبیل نام، نام فایل داده، نام فایل کارنامه (Log)، اندازه فایل داده و کارنامه را از طریق فرم دریافت می‌کند.
- ☒ ایجاد، حذف و ویرایش جداول به صورت پویا. اطلاعات جداول از قبیل نام، لیست فیلدها و نوع فیلدها را از طریق فرم دریافت می‌کند.

☒ ایجاد، حذف و ویرایش رویه‌های ذخیره شده به صورت پویا.

☒ ایجاد پرس‌وجو به صورت پویا. یعنی، نام فیلدهای گزارش، جداول تشکیل دهنده گزارش و شرط نمایش را به صورت پویا از کاربر دریافت می‌کند و اطلاعات بانک اطلاعاتی را نمایش می‌دهد. در هنگام تولید شرط می‌توان از عملگرهای or و AND برای ایجاد شرط‌های ترکیبی نیز استفاده کرد.

8. برنامه‌ای که در هنگام کار بر روی یک برنامه از آن پشتیبان تهیه می‌کند. در این برنامه کاربر می‌تواند اطلاعات زیر را انتخاب کند:

✖ انتخاب مسیر مبدا پشتیبان گیری یا انتخاب مسیر پیش فرض برای پشتیبان گیری.

✖ انتخاب مسیر مقصد پشتیبان گیری

✖ تعیین مدت زمانی که پس از آن مدت زمان هر بار باید پشتیبان تهیه شود.

البته در این برنامه یک کلاس به نام RequirementClass وجود دارد. این کلاس ماژول‌های از قبیل تنظیم و پرکردن DataGridView، ماژول دریافت مقدار سلول کلیک شده در DataGridView، ماژول تغییر زبان ورودی سیستم، ماژول فرمت بندی اعداد رقمی برای پول، ماژول تهیه لیست فرم‌های باز برنامه، ماژولی که اجرای مجدد برنامه را چک می‌کند. کلاس تقویم روز جلالی، کلاس تبدیل عدد به حروف، کلاس عملیات بر روی پایگاه داده و کلاس انتقال فرم با کلیک بر روی هر قسمت آن آماده است.

کتاب شامل 216 صفحه است  
که فایل الکترونیکی آن را می  
توانید از سایت کتابراه تهیه کنید.

<http://ktbr.ir/b۲۸۳۹۹>