

キー入力

キーボードイベント

通常、文字を入力する場合はすでに用意されているテキスト入力用のコントロールを使います
キー入力用のコントロールについては後ほど説明するとします

しかし、これ以外にも簡易処理などのために直接キー入力を受けつけたり
タイピングゲームなどのように、ウィンドウに直接入力してもらう必要がある場合があります
これを行うには、キーボードイベントを処理しなければなりません

キーボードのキーを押すと **Control.OnKeyDown()** メソッドと
登録されている **Control.KeyDown** イベントが呼び出されます
同様に、キーを離せば **Control.OnKeyUp()** メソッドと
登録されている **Control.KeyUp** イベントが呼び出されます

```
protected virtual void OnKeyDown(KeyEventArgs e);  
protected virtual void OnKeyUp(KeyEventArgs e);  
public event KeyEventHandler KeyDown;  
public event KeyEventHandler KeyUp;
```

e には、入力されたキーの情報をパックした **KeyEventArgs** オブジェクトが指定されます
これは **System.Windows.Forms.KeyEventArgs** クラスです

```
Object  
    EventArgs  
        KeyEventArgs
```

```
public class KeyEventArgs : EventArgs
```

このクラスのコンストラクタは次のようになっています

```
public KeyEventArgs(Keys keyData);
```

keyData には、入力されたキーを示す **Keys** 列挙型の値を指定します
この **System.Windows.Forms.Keys** 列挙型は、キーを示すために使われます

```
public enum Keys
```

この列挙型のメンバは、キーの数に合わせて数多く存在するため
この場では省略しますが、基本的には直感的に理解できる名前になっています
くわしくは、Microsoft のヘルプを参照してください

イベントが発生した時に押された、または離されたキーを知るには
KeyEventArgs.KeyCode または
KeyEventArgs.KeyData プロパティを使います
KeyCode はキーコードを、**KeyData** はキーのデータを返すと定義されています

```
public Keys KeyCode {get;}  
public Keys KeyData {get;}
```

これらのプロパティは、イベント発生時のキーを表す **Keys** の値を返します
この時の、ネイティブなキーコードを取得したい場合は
KeyEventArgs.KeyValue プロパティで得ることができます

```
public int KeyValue {get;}
```

通常、使う必要はないと思われますが
ネイティブな API とやりとりをする時や、特別にコードが必要な場合に使えるでしょう

Control クラスのイベントメンバを使ってキーイベントを処理する場合
KeyEventArgs.KeyValue.KeyEventHandler デリゲートを使います

```
public delegate void KeyEventHandler(  
    object sender , KeyEventArgs e  
);
```

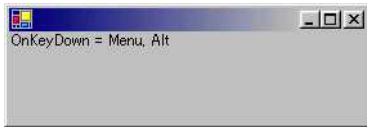
sender には、イベント発生元のオブジェクトが
e には発生時のキー情報を示す **KeyEventArgs** オブジェクトが指定されます

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    string key;  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        g.DrawString(key , Font , Brushes.Black , 0 , 0);  
    }  
    override protected void OnKeyDown(KeyEventArgs e) {  
        key = "OnKeyDown = " + e.KeyData;  
        Invalidate();  
    }  
    override protected void OnKeyUp(KeyEventArgs e) {
```

```

key = "OnKeyUp = " + e.KeyCode;
Invalidate();
}
}

```



このプログラムは、キーボードイベントが発生すると
 KeyEventArgs から Keys の値を取得して、その文字列表現を表示します

システムに密接に関連したキーは入力に対して重要な存在です
 すなわち、Alt キー、Shift キー、Control キーの3つです
 これらのキーに関しては、イベント発生時に押されているかどうかを調べる
 専用のプロパティが定義されているので、これを使うと便利でしょう

Alt キーは **KeyEventArgs.Alt** プロパティ
 Shift キーは **KeyEventArgs.Shift** プロパティ
 Control キーは **KeyEventArgs.Control** プロパティを使います

```

public virtual bool Alt {get;}
public virtual bool Shift {get;}
public bool Control {get;}

```

これらのプロパティは、キーが押されていれば true を
 キーが離されていれば false を返します

押されるキーそのものには興味が無く、単純に入力された文字を受け取りたい場合
 上のようなやり方では、Keys 列挙型と変換する文字のテーブルが必要です
 しかし、そのようなことをしなくても **Control.OnKeyPress()** メソッド
 または **Control.KeyPress** イベントを用いれば簡単に実現できます

```

protected virtual void OnKeyPress(KeyPressEventArgs e);
public event KeyPressEventHandler KeyPress;

```

e には、KeyPressEventArgs 型オブジェクトを指定します
 これは **System.Windows.Forms.KeyPressEventArgs** クラスです

```

Object
  EventArgs
    KeyPressEventArgs

```

```

public class KeyPressEventArgs : EventArgs

```

このクラスのコンストラクタは次のように定義されています

```

public KeyPressEventArgs(char keyChar);

```

keyChar には、入力されたキーの文字を指定します
 イベントが発生し、これを処理するメソッドは
KeyPressEventArgs.KeyChar プロパティを使ってこれを得られます

```

public char KeyChar {get;}

```

このプロパティを使えば、入力されたキーの文字を直接受け取ることができます
 コントロールが直接キーの文字を得る必要がある場合に使うことができますでしょう

Control のイベントメンバを使ってこれを処理する場合
System.Windows.Forms.KeyPressEventHandler デリゲートを使います

```

public delegate void KeyPressEventHandler(
    object sender , KeyPressEventArgs e
);

```

sender にはイベント発生もとのオブジェクトを
 e には入力されたキーの情報を保有する KeyPressEventArgs オブジェクトを指定します

```

using System.Windows.Forms;
using System.Drawing;

class WinMain : Form {
    string key;
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        g.DrawString(key , Font , Brushes.Black , 0 , 0);
    }
    override protected void OnKeyPress(KeyPressEventArgs e) {
        key = "OnKeyPress = " + e.KeyChar;
        Invalidate();
    }
}

```

このプログラムは、先ほどのプログラムを OnKeyPress() メソッドに変え
入カイベントに対して、入力されたキーの文字をウィンドウに描画するように変更しました

[前のページへ](#)

[戻る](#)

[次のページへ](#)