

プログラミング入門 (第2・3回)

ソフトウェア情報学研究科
分散システム学講座
2年 及川一樹

今日の予定

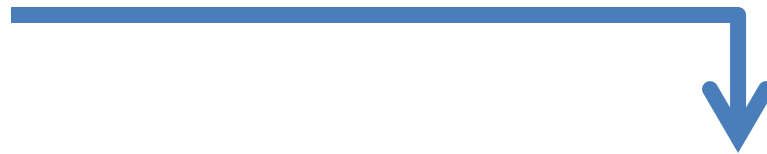
- ▶ 前回のおさらい
- ▶ if 文
- ▶ for文

前回のおさらい (1)

▶ イベントドリブンプログラミングモデル



クリック!



```
private void button1_Click (object sender, EventArgs e)
{
    // 波括弧に囲まれた、この場所に記述されたコードが実行される
}
```

- ▶ プロパティ: オブジェクトの振る舞いを変える
 - » Textプロパティ: 表示される文字列を取得・変更する
 - » ForeColorプロパティ: 文字色を取得・変更する
- ▶ メソッド: オブジェクトに命令を行う
 - » int.Parseメソッド: 文字列を整数に変換する命令
 - » float.Parseメソッド: 文字列を小数に変換する命令

前回のおさらい (2)

▶ 変数

- » 数学における変数と同じ意味
- » 何らかの値を保存しておく記号

▶ 演算式

- » 変数や数字、演算子を用いて演算を行う式

```
int x = 10;  
int y = 20;  
int z = x + y + 2;
```

```
float height = 1.72;  
float weight = 65.8;  
float bmi = weight / (height * height);
```

if 文 (条件分岐)

- ▶ 条件式によって条件を評価し、結果によって処理を分岐する

```
» if (条件式) {  
    // 条件式が成り立つときの処理  
} else {  
    // 条件式が成り立たないときの処理  
}
```

```
int x = 10;  
if (x < 10) {  
    MessageBox.Show (“xは10未満です”);  
} else if (x == 10) {  
    MessageBox.Show (“xは10です”);  
} else {  
    MessageBox.Show (“xは10よりも大きいです”);  
}
```

条件式

- ▶ 基本的には右辺と左辺の比較
 - » $x == y$: x と y が等しいかどうか評価
 - » $x < y$: x よりも y が小さいかどうか評価
 - » $x > y$: x よりも y が大きいかどうか評価
 - » $x <= y$: x が y 以下かどうか評価
 - » $x >= y$: x が y 以上かどうか評価
- ▶ 複数の式を組み合わせてすることも出来る

```
int x = 10;
if ((x >= 10) && (x < 20)) {
    MessageBox.Show (“xは10以上20未満です”);
} else if ((x >= 20) || (x < 10)) {
    MessageBox.Show (“xは20以上か、10未満です”);
}
```

演習1

- ▶ 一番小さい値を出力するプログラムを作ろう
 - » テキストボックスを3つ、ボタンを1つ配置
 - » テキストボックスに入力された3つの値を if文を使って比較し、一番小さい値をMessageBoxで表示させる

- ▶ カンペ
 - » テキストボックスの値を整数に変換する方法

```
int value = int.Parse (textBox1.Text);
```

- » MessageBoxで数値を表示させる方法

```
MessageBox.Show (value.ToString ());
```

演習2

- ▶ 前回作ったBMIを求めるプログラムを改良し、BMI値に基づいて“やせ”、“標準”、“肥満”を判定できるようにする
- ▶ BMI
 - » 18.5未満 → やせ
 - » 18.5～25未満 → 標準
 - » 25以上 → 肥満

for 文

- ▶ 指定回数だけ、処理を繰り返す（厳密には違いますが・・・）
- ▶ 例：1～10の値を全て加算する

```
int sum = 0;
for (int i = 1; i <= 10; i ++ ) {
    sum = sum + i;
}
MessageBox.Show (sum.ToString ());
```

- ▶ 上記例で利用されている変数 i はループカウンタと呼ばれ、この例の場合、ループのたびに1から1つずつ増加していき、 $i \leq 10$ という条件式が成り立たなくなるとループから抜ける

演習3

- ▶ 入力された数値の階乗を求めるプログラムを作成しよう

演習4

▶ 入力された値が素数かどうかチェックするプログラムを作ろう

▶ 判定方法

» エラトステネスの篩い

- ▶ 素数は1か自身の値でしか割りきれない
- ▶ つまり、2～(自身の値-1) で割り切れるかチェックし、割り切れる値が存在したら合成数、存在しなかったら素数と判定できる
- ▶ 割り切る → 剰りが0。 余りを求める演算子 % を使う

» 余裕がある人

- ▶ フェルマーテストやラビンミラー素数判定法を調べて作ってみよう
- ▶ 乱数の発生方法

```
Random random = new Random ();
```

← 上の方で宣言

```
int r = random.Next (min,max);
```

← min以上max未満の
乱数を生成するとき。