

プログラミング入門 (第1回)

ソフトウェア情報学研究科
分散システム学講座
1年 及川一樹

目次

1. はじめに
 1. 自己紹介
 2. プログラミング入門に関して
2. プログラミング入門 (第1回)
 1. C#について
 2. コメント
 3. Hello World !
 4. 入力と出力
 5. 変数と型
 6. 式
 7. 評価式
 8. if文
 9. for文
3. おわりに

自己紹介

名前: 及川一樹

学年: ぴっかっぴかの1年生

(但し、大学院博士前期課程)

講座: 分散システム学講座 (高田2研)

住処: ソフトウェア情報学部B棟 講座研究室5

趣味: プログラミング、音楽鑑賞、読書、ピアノ

好きな言語: C#, Python, XSL

好きなディストリビューション: Gentoo Linux

好きな食べ物: りんご

プログラミング入門に関して

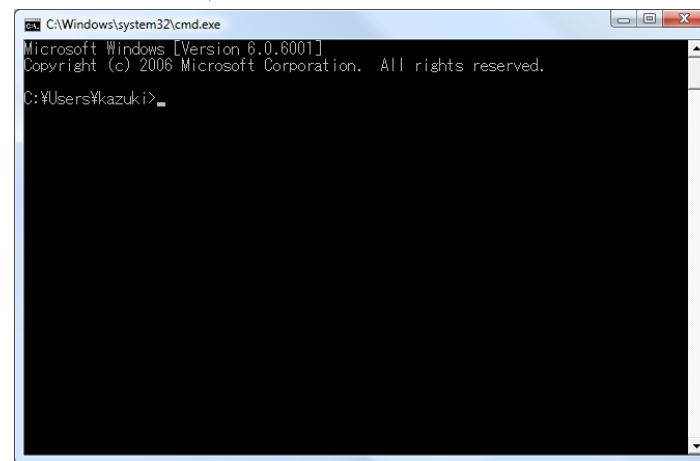
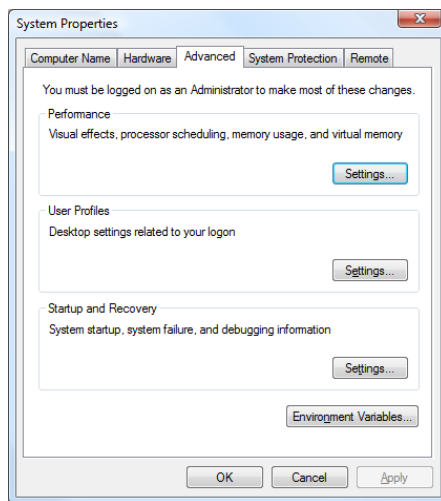
- Windowsプログラミングを体験して貰います
 - » 一回目: 制御構文
 - » 二回目: GUIで遊ぶ
 - » 三回目: 総仕上げ
- プログラミングの楽しさが分かって貰えばなあ...

C#について

- 今回は、C# という言語を利用します
 - » Microsoftが開発した言語
 - » 簡単で使いやすく、書き方もC言語やJavaに似ている
 - » 残念なことに、ワークステーションではC#は使えません

Hello World ! (1)

- 一般的に、最も単純なプログラムとして挙げられる
- 今回は最初なので、GUI ではなく CUI を利用します



1. Visual Studio を起動する
2. Visual C# → Windows → コンソールアプリケーション を選択
3. プロジェクト名に HelloWorld と入力
4. OKボタンをクリック

Hello World ! (2)

- `static void Main (string[] args)`
`{`
`}`
 - » プログラムが起動したときに、呼び出される場所
 - » この中に、実行したい処理を記述する
- `Console.WriteLine ("Hello World !");`
 - » Console (CUIの黒いウィンドウ) に、"Hello World !" と1行書き込み (`WriteLine`) という指示をしている
 - » このように、1つの指示を"文"と呼び、行末に";"をつける
 - 逆に言うと、";"で終端するまでが"文"
- メニュー → デバッグ → デバッグなしで開始 を選択

コメント

- C#では
 - » // で始まる行
 - » /* と */ で囲まれた領域
- に書かれたプログラムを実行しません
- なので、そこにコメントとかを自由にかける！
- 例:
// おはよー
Console.WriteLine ("Hello World !");
// TODO: 続きを書く

入力と出力

- 出力
 - » “Hello World !” が表示されたよね？
 - » あれが、出力
- 入力
 - » さっき出てきた、黒い画面ではキーボードだけ使える
 - CUIではマウスは使えない
 - » キーボードからの入力を受け取るのが入力
 - ここでは、`Console.ReadLine()` を使います
- 以下の2行を追加してみよう
 1. `string input = Console.ReadLine ();`
 2. `Console.WriteLine (input);`

変数と型 (1)

- 型: データの種類を表す
 - » 数値型
 - 整数型
 - ☆byte型 (8bit, 符号無し, 0~255)
 - ☆short型 (16bit, 符号あり, -32,768 ~ 32,767)
 - ☆int型 (32bit, 符号あり, -2,147,483,648 ~ 2,147,483,647)
 - 浮動小数点型
 - ☆float型 (32bit, 有効桁数7桁)
 - ☆double型 (64bit, 有効桁数16桁)
 - » 真偽型
 - bool型 (true または false)
 - » 文字列型
 - string型

変数と型 (2)

- 変数
 - » 計算などの途中結果を保存しておく場所
 - » 型名 変数名; or 型名 変数名 = 初期値; という構文で**宣言**
 - `int age;`
 - `int age = 21;`
 - `double pi = 3.14159265;`
 - `string name = "及川一樹";`
 - » 変数を使う前には、宣言を行う必要がある
- 問題です
 1. 1~10までの数の足し算を行わせたいです。
何型を使えばいいですか？
 2. $1 \div 7$ という循環小数の循環節を見たいです。
何型を使えばいいですか？

式 (1)

- 代入式 (数学と同じ)
 - » `x = 12;`
 - xに12を代入
 - » `y = x;`
 - yにxの値を代入
- 二項演算 (数学と同じ)
 - » 加算 → +
 - » 減算 → -
 - » 乗算 → *
 - » 除算 → /
 - » 除算の余り → %
- `x = 3.0 / 7.0;`
 - » xに $3 \div 7$ の結果を代入
- 単項演算
 - » `x++;`
 - » `x--;`
 - 変数xの値を1増加/減少させる
 - 以下のプログラムで説明

```
int x = 10;  
x ++;  
Console.WriteLine (x);
```
- 代入演算
 - » 左辺の値と右辺の値を演算後、左辺に代入するという特殊な演算
 - » `x += 10;`

やってみよう (1)

1. その1 (prog1)

1. int型の変数x,y を利用
2. xは初期値5で初期化
3. yはxに3を加えた値を代入
4. xとyの値を出力

2. その2 (prog2)

- » キーボードから入力した数値を二倍した値を出力

やってみよう (2)

- その2の問題点
 - » `Console.ReadLine();` で読み取れるのは、文字列
 - » 文字列を二倍することは出来ない
- 解決するには
 - » 文字列を数値型に変換する必要がある
- 解決策
 - » `int.Parse (string);`
 - » `double.Parse (string);`
 - 括弧の中に指定した文字列を、数値型に変換する
 - 数値じゃない文字を指定するとエラーになるけど、この講義ではそういういぢわるは禁止です！ (笑)

評価式

- 条件をチェックして、成り立っていれば真 (true), 成り立たなければ偽 (false) を返す式のこと
 - » true/false を返すので、左辺には bool型 の変数を利用
- 比較演算子
 - » ==: 左右の値が等しいかをチェックする (例: $x == y$)
 - !=: 等しくない事をチェックする比較演算子
 - » <: 左右の値の大小関係をチェックする (例: $x < y$)
 - 他にも、>, <=, >= があるよん
- 試してみよう (prog3)
 1. `bool ret = 1 == 3;`
 2. `Console.WriteLine (ret);`

if文

- 評価式を利用し、その結果によって実行する処理を切り替える文
- if (評価式) {
 // 評価式が true の時に実行される場所
} else {
 // 評価式が false の時に実行される場所
}
- やってみよう (prog4)
 1. キーボードから数値を入力
 2. 数値が偶数なら、“偶数” と出力
 3. 数値が偶数でない場合は、“奇数” と出力
 - » ヒント
 - 2で割ったあまりが0なら偶数、それ以外なら奇数だよ

for文 (1)

- 指定した回数同じ処理を繰り返す文
- ```
for (int i=0; i < 10; i ++) {
 // 10回繰り返す文
}
```
- 変数*i* はループカウンタと呼び、繰り返しの度に変化していく
  - » `int i = 0;` と宣言されているので、初期値は0
    - もちろん、ここを変更すれば、初期値を10にすることもできる
  - » `i++` とあるので、繰り返しの度に 1増加していく
  - » `i++` を `i+=2` とすれば、繰り返しの度に 2増加するようになる
- `i < 10` という評価式が成立している間繰り返される
  - » 評価式が偽になれば繰り返りは終了し、for文の次の文に移る
  - » `i < 10` という部分は評価式であれば、どんな評価式でもOK
  - » 評価式を間違えると、無限に繰り返してしまい、プログラムが止まらないことも

## for文 (2)

- やってみよう (prog5)
  - » 1～10までの数値を出力し、その合計を最後に出力するプログラム
  - » 出力例:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
55

## for文 (3)

- やってみよう (prog6)
  1. キーボードから数値を入力
  2. その数値以下の数で、3の倍数を出力
  3. 最後に、3の倍数の数と、その和を出力出力例 (入力が12):

3

6

9

12

4

30