

第1章「stringを使う」

主なトピック:

- スtring型(文字列型) `std::string`の使い方
- コンソール(`std::cin`, `std::cout`)の使い方

2番目のソース

```
/*  
    greetings.cc  
    概要：ユーザに名前を聞いて、あいさつをするプログラム  
    Author: Keitaro Naruse  
    Date: 2021-10-02  
*/  
  
#include <iostream>  
#include <string>  
  
int main()  
{  
    // ユーザに名前を聞く  
    std::cout << "Please input your name:";
```

```
// 名前を読み込む
std::string name;    // nameの定義
std::cin >> name;    // nameに読み込み

// あいさつを書く
std::cout << "Hello, " << name << "!" << std::endl;

return 0;
}
```

このプログラムを実行すると

- 画面に
 - Please input your name:
- Vladimirと入力したとすると
 - Vladimir
- 画面には下が出力される
 - Hello, Vladimir!

string型(1/3)

- C言語では文字型 `char`
 - 文字列を扱うときは `char`型の配列として表現
 - `char name[80];`
 - 上の場合だと最大80文字
- C++言語では文字列型 `string`型
 - `std::string name;`
 - `name = "Vladimir";`
 - 何文字であっても構わない

string型(2/3)

- `#include <string>`
 - string型を利用するためのインクルード文
- `std::string name;`
 - string型の変数nameを宣言
- `std::cin >> name;`
 - 標準入力(std::cin)から文字列を読み込み, nameに格納

string型(3/3)

- 文字列の長さを調べるにはsize()
 - string型のメンバ関数
 - nameの中身が”Keitaro Naruse”だとして
 - name.size()とすると14が返る
- stringの初期化
 - std::string name1(“Keitaro Naruse”);
 - std::string name2 = “Keitaro Naruse”;
 - std::string name3 = name1;
 - std::string name4(name1);
 - name1からname4の中身は”Keitaro Naruse”
 - std::string spaces(8, ‘ ‘);
 - 8文字の*（アスタリスク）からなる文字列

標準入力ストリームと標準出力ストリーム

- 標準入力ストリーム(キーボード) `std::cin`
 - `std::cin >> first >> second;`
 - C言語では
 - `char first[80], second[80];`
 - `scanf("%s", first); scanf("%s", second);`
- 標準出力ストリーム(画面) `std::cout`
 - `std::cout << first << " " << second << std::endl;`
 - C言語では
 - `printf ("%s %s\n", first, second);`

型とオブジェクト

- 型: 値と演算を定義する
(例) int, double
- オブジェクト: 特定の型の値を保持するメモリ
(例) int a, double b;
- 値: 型に従って解釈されるビット列
(例) a = 0, b = 0.0;
- 変数: 名前付きのオブジェクト
(例) a, b

整数型

- `int` 型は整数の値と演算を定義する
- `int a = 7, b = 9, c;`
 - この`=`は初期化演算子
 - 変数の値は指定された整数値
- `c = a + b;`
 - この`=`は代入演算子
 - `+` は加算の演算子
 - `c`の値は16

文字列型

- string 型は文字列の値と演算を定義する
- string a = "alpha", b = "beta", c;
 - この=は初期化演算子
 - 変数の値は指定された文字列
- c = a + b;
 - この=は代入演算子
 - + は文字列連結の演算子
 - cの値は"alphabet"

サンプルコード

- `std::string name;`
- `std::cin >> name; // "Vladimir"が入力されたとして`
- `const std::string greeting = "Hello, " + name + "!";`
 - `greeting`は”Hello, Vladimir!”になる
 - `const`宣言はプログラムの中で変数(この場合は`greeting`)の内容を変更させない
 - `const` 宣言したときは, 必ず値を与える
 - `name`は変数宣言した後で値が変わる(代入される)から`const`宣言できない

サンプルコード

- `const std::string spaces(greeting.size(), ' ');`
 - spaces というstring型の変数は,
 - greetingの文字列と同じ長さ(16文字)で, 空白だけからなる文字列で初期化
- `const std::string second = “* ” + spaces + “ *”;`
 - second の中身は”* *”
 - 全体で20文字(2+16+2)
- `const std::string first (second.size(), '*');`
 - first の中身は”*****” (20文字)