

情報システムプログラミングⅡ (3回目)

2024年4月26日 (金)

3～4限

授業内容

- 講義内容（教科書の204～214ページ + α ）
 - 構造体の注意点
 - 三項演算子
 - Pythonにおける構造体と三項演算子
- 演習課題

構造体の注意点

■構造体に利用できる演算子

- 代入以外は利用できない（メンバに対しては利用できる）

操作	表示 printf() など	計算 +、-、*、/ など	比較 ==、!=、< など	代入 (コピー) =
基本型 ※	○	○	○	○
構造体型	×	×	×	○

※ int 型、char 型など。なお、String 型は基本型には含まれない。

構造体の注意点

■ 構造体の表示

- 構造体の情報をまとめて表示することはできない

```
Monster suzaku = {"朱雀", 100, 30};
```



```
printf("%s", suzaku);
```

朱雀 (HP=100/ 攻撃 =30) と表示したい

- 各メンバごとに表示する必要がある

```
const String TEMPLATE = "%s : HP=%3d 攻撃力=%2d\n";  
printf(TEMPLATE, seiryu.name, seiryu.hp, seiryu.attack);  
printf(TEMPLATE, suzaku.name, suzaku.hp, suzaku.attack);  
printf(TEMPLATE, byakko.name, byakko.hp, byakko.attack);  
printf(TEMPLATE, genbu.name, genbu.hp, genbu.attack);
```


構造体の注意点

■ 構造体の計算

- 各メンバごとに計算する必要がある

```
int main(void)
{
    typedef struct {
        int x;
        int y;
    } Point;

    Point p1 = {50, 70};
    p1++;
}
```

ある座標を表す構造体を定義

x=51 y=71 としたいが、エラー



```
p1.x++;
p1.y++;
```

構造体の注意点

■ 構造体の比較

- 各メンバごとに比較する必要がある

```
Monster suzaku1 = {"朱雀", 100, 30};  
Monster suzaku2 = {"朱雀", 105, 25};  
:  
if (suzaku1 == suzaku2) { }  
:  
}
```

✖ コンパイルエラー

■ 構造体の代入（コピー）

- 同じ型の構造体であればまとめて代入できる

```
Monster suzaku = {"朱雀", 100, 30};  
Monster suzaku2;  
suzaku2 = suzaku;
```

○ 3つのメンバの内容がすべてコピーされる

三項演算子

■三項演算子とは3つのオペランドを持つ演算子

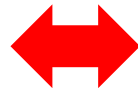
- 条件演算子, 三項条件演算子ともいう
- 三項演算子を利用するための構文

条件式 ? 値 1 : 値 2

※条件式が満たされれば値 1 に、そうでなければ値 2 に全体が化ける。

- 分岐処理を簡単に記述できる

```
if (a == 'x') printf("OK\n");  
else printf("NG\n");
```



```
printf("%s\n", a == 'x' ? "OK" : "NG");
```

Pythonにおける構造体と三項演算子

■Pythonの構造体

- Pythonには構造体がない（似たような仕組みは実装できる）
 - クラス（class）で代用できる

■Pythonの三項演算子

- Pythonで三項演算子を利用するための構文

条件式が真のときの式 if 条件式 else 条件式が偽のときの式



```
if 条件式:  
    条件式が真のときの式  
else:  
    条件式が偽のときの式
```