

オブジェクト指向と ゲームプログラミング

基礎編 - 第6回 多分岐

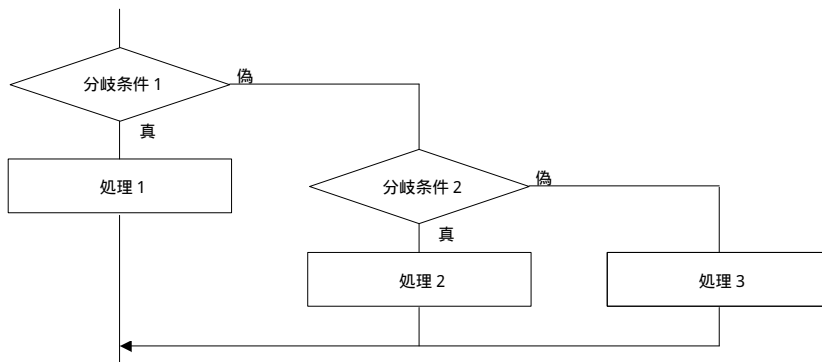
else-ifによる多分岐

if-else文を連ねて条件分岐を多重に行う場合、else-if文を使用することができます。たとえば、

```
if(条件式 1) {  
    条件式 1 を満たしたときに実行する処理  
} else {  
    if(条件式 2) {  
        条件式 2 を満たしたときに実行する処理  
    } else {  
        if(条件式 3) {  
            条件式 3 を満たしたときに実行する処理  
        } else {  
            条件式 3 を満たさないときに実行する処理  
        }  
    }  
}
```

というようなif-else文は、以下のようにelse-if文で記述することができます。

```
if(条件式 1) {  
    条件式 1 を満たしたときに実行する処理  
} else if(条件式 2) {  
    条件式 2 を満たしたときに実行する処理  
} else if(条件式 3) {  
    条件式 3 を満たしたときに実行する処理  
} else {  
    条件式 3 を満たさないときに実行する処理  
}
```



if-else-if

switch-caseによる多分岐

switch-case文は、変数の値によって実行する処理を分岐させることができます。switch-case文では、

変数の値が、
値 1 なら処理 1 を実行し、
値 2 なら処理 2 を実行し、
値 3 なら処理 3 を実行し、
値 4 なら処理 4 を実行し、
上記のどれでもなければ処理 5 を実行する

という分岐をさせることができます。switch-case文の書式は次のようになっています。

```

switch(変数) {
case 値 1 または定数式 1 :
    処理 1
    break;

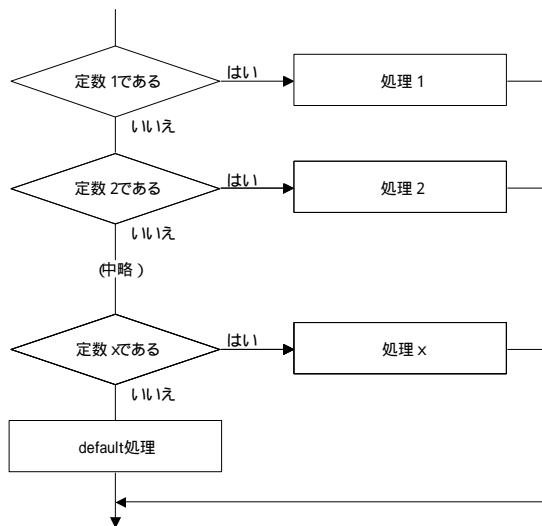
case 値 2 または定数式 2 :
    処理 2
    break;

(中略)

case 定数式 x :
    処理 x
    break;

default:
    すべてのcaseを満たさなかったときの処理
    break;
}

```



case文とdefault文の最後は":" (コロン) で終了します。case文の分岐条件に記述できるのは整数型の定数式のみです。default文は必要のないときは省略することができます。また、case文の最後にあるbreakは、switch文を抜けるために必要です。breakがないと、breakが見つかるまで、またはswitch文の最後まで処理が実行されます。

```

switch(value) {
case 1:
case 2:
case 3:
    処理 1
    break;

case 4:
    処理 2
    break;

default:
    処理 3
    break;
}

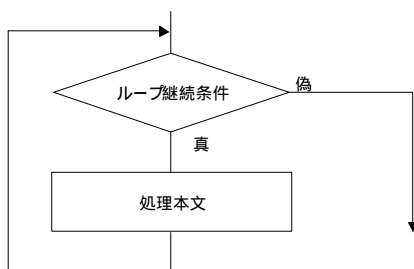
```

このような場合は、変数valueの値が1、2、3の場合は処理 1 が、4の場合は処理 2 が、それ以外の場合は処理 3 が実行されます。

switch-case文では、整数型の値による分岐しかできません。小数型や文字列による分岐、複雑な条件で分岐先が多い場合は、if-else文を連ねて分岐させます。

whileによる反復

while文は、以下のように処理が行われます。ループに入る前に条件判定を行い、条件が成り立つ間処理を反復します。



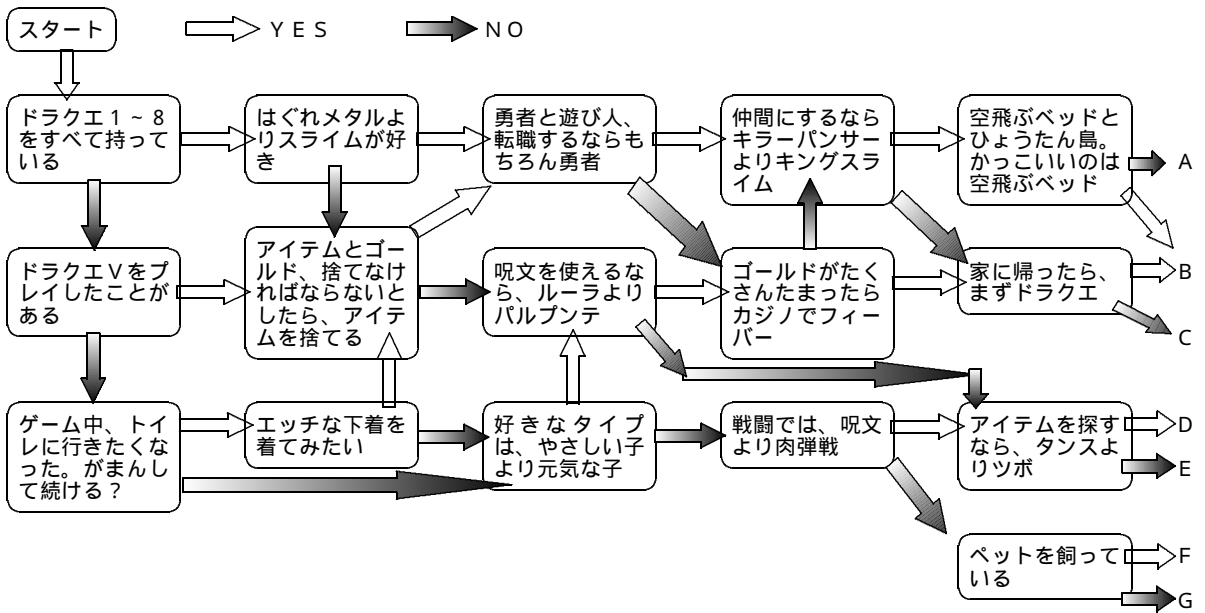
while文の書式は以下のようになります。

```
while(ループ継続条件) {
    ...
    処理本文
    ...
}
```

「ループ継続条件」は、ループを継続させるための条件を記述します。この条件が真の間、ループが継続されます。ループに入る前に判定が行われるので、条件によってはループに入らない場合があります。

練習問題

メッセージボックスを用いて、以下のような相性診断を行うプログラムを作成しましょう。



- A...主人公 主人公にたどり着いたあなたは、行動力大。ひとりでがんばれ
- B...パーバラ 元気で明るいパーバラが、気弱なあなたををひっぱってくれるでしょう
- C...チャモロ いつもまじめなあなたは、チャモロと一緒に呪文を研究しましょう
- D...ハッサン 自由気ままなハッサンが、あなたの力強い仲間になってくれるでしょう
- E...テリー クールなテリーと良い友達になれるでしょう
- F...ミレーユ やさしいお姉さんのミレーユに甘えてみたい、それが本心ようです
- G...ファルシオン 決して戦わない、白馬のファルシオンと心が通じそう

ヒント

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
    int no = 1; // 質問No;
```

ヒント 多分岐 反復

ヒント から の質問の中から変数 で指定される質問を選ぶ。回答をもとに次の質問を変数 に設定する。これをゴールまで す。

オブジェクト指向と ゲームプログラミング

番外編 2進数

2進数

2進数とは、2を基数として数値を表現する方法のことです。コンピュータでは、2進数が使われていて、0と1の2種類だけで数値を表します。電圧が高ければ1、低ければ0と決めておき、複数の信号の組み合わせで数値を表現しているのです。

この2進数の1桁は、コンピュータが扱う情報の最小単位で、ビット(bit:binary digit)と呼ばれます。たとえば、8ビットとは、2進数8桁であることを表し、256通りの情報量を持ちます。16ビットでは、2進数16桁となり、65,536通りの情報量を持ちます。一般に、nビットの情報量は、2のn乗通りの情報を表現することができます。

2進数から10進数の変換

普段使用している10進数を考えてみます。10進数で893とは、「8百9+3」という意味になります。10進数の各桁にそれぞれ 10^2 、 10^1 、 10^0 という重みがあるからです。893は、

$$8 \times 10^2 + 9 \times 10^1 + 3 \times 10^0 = 893$$

となります。10進数の場合、1桁左に書かれた数字は、1桁右の数字よりも10倍の重みを持っています。この数字を基数といいます。10進数の基数は当然10です。

2進数でも同様に、各桁にそれぞれ重みがあり、1桁左に書かれた数字は、1桁右の数字よりも2倍の重みを持っています。2進数の基数は2だからです。たとえば、2進数の「1001」は、

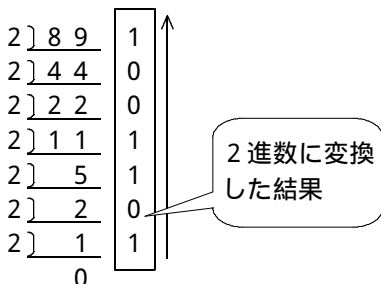
$$\begin{aligned} & 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ = & 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 \\ = & 9 \text{ (10進数)} \end{aligned}$$

となります。各けたの重みをかけたものの合計が10進数での数値になります。

10進数から2進数の変換

10進数を2進数に変換するには、変換したい10進数を商が0になるまで2で割りつづけ、商と余りを求めます。

10進数の「89」を2進数に変換してみましょう。



となり、10進数の89は、2進数では「1011001」です。

練習問題

以下の数値を2進数なら10進数に、10進数なら8桁の2進数に変換しましょう。

- | | |
|-----------------------------|-------------------------|
| (1) 11111111 ₍₂₎ | (5) 14 ₍₁₀₎ |
| (2) 10000000 ₍₂₎ | (6) 31 ₍₁₀₎ |
| (3) 10101010 ₍₂₎ | (7) 47 ₍₁₀₎ |
| (4) 01010101 ₍₂₎ | (8) 226 ₍₁₀₎ |