

# プログラミング1

Proguraming1

## ★疑似言語用いたプログラミングの学習

プログラミングの学習方法として、疑似言語を使用します。

[疑似言語の記述形式] 変数と注釈

| 記述形式  | 説明                            |
|---|-------------------------------|
| ○ <u>手続名</u> 又は <u>関数名</u>                  | 手続又は関数を宣言する。                  |
| <u>型名</u> : <u>変数名</u>                      | 変数を宣言する。                      |
| /* <u>注釈</u> */                             | 注釈を記述する。                      |
| // <u>注釈</u>                                |                               |
| <u>変数名</u> ← <u>式</u>                       | 変数に <u>式</u> の値を代入する。         |
| <u>手続名</u> 又は <u>関数名</u> ( <u>引数</u> , ...) | 手続又は関数を呼び出し, <u>引数</u> を受け渡す。 |

## 分岐

|   |  |
|---|--|
| <pre>if (<u>条件式 1</u>)<br/>    <u>処理 1</u><br/>elseif (<u>条件式 2</u>)<br/>    <u>処理 2</u><br/>elseif (<u>条件式 n</u>)<br/>    <u>処理 n</u><br/>else<br/>    <u>処理 n + 1</u><br/>endif</pre> | <p>選択処理を示す。</p> <p><u>条件式</u>を上から評価し、最初に真になった<u>条件式</u>に対応する<u>処理</u>を実行する。以降の<u>条件式</u>は評価せず、対応する<u>処理</u>も実行しない。どの<u>条件式</u>も真にならないときは、<u>処理 n + 1</u>を実行する。</p> <p>各<u>処理</u>は、0以上の文の集まりである。</p> <p>elseif と<u>処理</u>の組みは、複数記述することがあり、省略することもある。</p> <p>else と<u>処理 n + 1</u>の組みは一つだけ記述し、省略することもある。</p> |
|---|--|

## 反復

|   |   |
|---|---|
| <code>while</code> ( <u>条件式</u> )<br><u>処理</u><br><code>endwhile</code> | 前判定繰返し処理を示す。<br><u>条件式</u> が真の間、 <u>処理</u> を繰返し実行する。<br><u>処理</u> は、0以上の文の集まりである。                 |
| <code>do</code><br><u>処理</u><br><code>while</code> ( <u>条件式</u> )       | 後判定繰返し処理を示す。<br><u>処理</u> を実行し、 <u>条件式</u> が真の間、 <u>処理</u> を繰返し実行する。<br><u>処理</u> は、0以上の文の集まりである。 |
| <code>for</code> ( <u>制御記述</u> )<br><u>処理</u><br><code>endfor</code>    | 繰返し処理を示す。<br><u>制御記述</u> の内容に基づいて、 <u>処理</u> を繰返し実行する。<br><u>処理</u> は、0以上の文の集まりである。               |

### <Ex> プログラミングの基本要素

x と y を正の整数とするとき、次のプログラムを実行した結果の戻り値 q、r として適切なものを解答群の中から選りなさい。ただし、正の整数 x と y には 0 を含めない。

#### プログラム

```
○整数型 : syo(整数型 : x, 整数型 : y)
  整数型 : q, r
  q ← 0
  r ← x
  for (rがy以上)
    r ← r - y
    q ← q + 1
  endfor
  return q, r
```

#### 解答群

|   | q の値           | r の値           |
|---|----------------|----------------|
| ア | $x \div y$ の余り | $x \div y$ の商  |
| イ | $x \div y$ の商  | $x \div y$ の余り |
| ウ | $y \div x$ の余り | $y \div x$ の商  |
| エ | $y \div x$ の商  | $y \div x$ の余り |

<Q>関数calcMeanは、要素数が1以上の配列dataArrayを引数として受け取り、要素の値の平均を戻り値として返す。プログラムの中のa、bに入る字句の適切な組合せを選びなさい。ただし、配列の要素番号は1から始まる。

```
○実数型 : calcMean(実数型の配列 : dataArray) /*関数の宣言*/
  実数型 : sum, mean
  整数型 : i
  sum ← 0
  for(iを1からdataArrayの要素数まで1ずつ増やす)
    sum ← 
  endfor
  mean ← sum ÷  /*実数として計算する*/
  return mean
```

- |   |                        |                       |
|---|------------------------|-----------------------|
| ア | a : sum + dataArray[i] | b : dataArrayの要素数     |
| イ | a : sum + dataArray[i] | b : (dataArrayの要素数-1) |
| ウ | a : sum - dataArray[i] | b : dataArrayの要素数     |
| エ | a : sum - dataArray[i] | b : (dataArrayの要素数-1) |



<Q> ある学校の成績評価は、90点以上では“A”、70点以上90点未満では“B”、60点以上70点未満では“C”、60点未満では“D”である。この学校の試験の点数は整数値である。点数の範囲は0～100点とする。プログラムの関数 seiseki は、試験の点数を引数として受け取り、成績評価A～Dを戻す。なお、不適切な点数の処理は行わない。

○整数型 : seiseki(整数型 : ten)

文字型 : hyouka

if(tenが90点以上)

hyouka ← “A”

hyouka ← “B”

elseif(tenが60点以上)

hyouka ← “C”

else

hyouka ← “D”

endif

return hyouka

ア elseif(tenが90点未満)

イ else

ウ elseif(tenが70点以上)

エ elseif(tenが60点以上)





<Q> 次のプログラムは1からN ( $N \geq 1$ ) までの整数の総和を求め、結果を変数 z に入れる。

```
○整数型 : sum(整数型 : N)
  整数型 : i, x
  x ← 0
  i ← 1
  while(  )
    x ← x + i
    i ← i + 1
  endwhile
  return x
```

- ア iがNよりも小さい
- イ iがNよりも大きい
- ウ iがNと等しい
- エ iがN以上
- オ iがN以下
- カ xがN以下



<Q> 職業性ストレス簡易調査票の回答結果から高ストレス者を判定するプログラムを示す。変数resultには初期値0を格納しておき、高ストレス者を判定したらresultに1を格納する。なお高ストレス者の判定条件は以下の2つである。

- ① 領域Bの合計点が77点以上である。
- ② 領域Bの合計点が63点以上76点以下であり、かつ領域AとCの合計点の和が76点以上である。

```
○stress(整数型 : a, 整数型 : b, 整数型 : c)
  整数型 : result /*判定結果*/
  整数型 : total
  result ← 0
  total ← b /*領域Bの合計点*/
  if(totalが77点以上)
    result ← 1
  elseif(  )
    total ← a + c /*領域Aと領域Cの合計点の和*/
    if(  )
      result ← 1
    endif
  endif
endif
```

|   |                 |                 |
|---|-----------------|-----------------|
| ア | a : totalが63点未満 | b : totalが77点以上 |
| イ | a : totalが63点未満 | b : totalが77点以下 |
| ウ | a : totalが63点以上 | b : totalが75点以下 |
| エ | a : totalが63点以上 | b : totalが76点以上 |

