

本サンプル問題の著作権は日本商工会議所に帰属します。

また、本サンプル問題の無断転載、無断営利利用を厳禁します。本サンプル問題の内容や解答等に関するお問い合わせは、受け付けておりませんので、ご了承ください。

## 日商プログラミング検定 EXPERT (VBA) サンプル問題

### 知識科目

第1問 (知識4択: 20問)

1. 実行時エラーが発生してマクロを停止したとき、マクロをリセットするためにクリックするアイコンは、次のうちどれか。最もふさわしいものを選択しなさい。

①



②



③



④



2. 以下の各要素のうち、プロパティでないものはどれか。次から選びなさい。

```
ActiveCell.Offset(0, 1).Font.Color = 10
```

① ActiveCell

② Offset

- ③ Font
- ④ Color

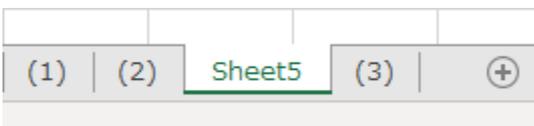
3. シートの状態が図のようなとき、次の文を実行した。実行結果として、もっともふさわしいものはどれか。  
Worksheets.Add before:=Worksheets(2)



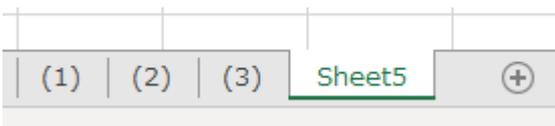
①



②



③



④



4. 次の文により指定されるセルは次のうちどれか。最もふさわしいものを選択しなさい。  
Cells(2, 3).Offset(0, 1)

- ① D2
- ② C3
- ③ B4
- ④ C2

5. 以下のように宣言された変数 A に代入する文として、最もふさわしいものを選択しなさい。  
Dim A As Range

- ① Set A = Range("A1")
- ② A = Range("A1")
- ③ A = 10
- ④ set A = 10

6. 次の文を実行した結果、変数 B が True となる A の値はどれか。最もふさわしいものを選択しなさい。

$B = A > 10 \text{ And } A < 20$

- ① 15
- ② 25
- ③ 5
- ④ 該当する値は存在しない

7. String 型の変数 s に文字列が格納されているとき、最初の 1 文字を削除する文は次のうちどれか、もっともふさわしいものを選びなさい。

例)  $s = \text{"ABCD"}$  のとき、実行結果が  $s = \text{"BCD"}$  となる。

- ①  $s = \text{Right}(s, \text{Len}(s) - 1)$
- ②  $s = \text{Right}(s, \text{Len}(s))$
- ③  $s = \text{Left}(s, \text{Len}(s) - 1)$
- ④  $s = \text{Left}(s, \text{Len}(s))$

8. 固定長配列の添字の下限に関する説明について、最もふさわしいものを選択しなさい。

- ① 配列を宣言するとき、任意の整数で指定をすることができる。
- ② 配列の添字の最低値は 0 であり、変更することはできない。
- ③ 配列の添字の最低値は 1 であり、変更することはできない。
- ④ 配列の添字の最低値は、宣言セクションで任意の整数を指定することができる。

9. Integer 型の配列 data の各要素が下図のようであるとき、下の文を実行した。実行後の C4 セルの値は次のうちどれか。選びなさい。

$\text{Range}(\text{Cells}(4, 2), \text{Cells}(6, 3)) = \text{data}$

	1	2
1	195	109
2	151	74
3	137	75

- ① B5
- ② C4
- ③ A2
- ④ B1

10. ソートが下図のようであるとき、次のプログラム片において、処理1が実行されるのは、ActiveCellがどのセルであるときか。次から選びなさい。

	A	
1	10	If Not IsNumeric(ActiveCell) Then 処理 1
2	3.14	Else
3	3,000	処理 2
4	3年	End If
5		

- ① A4
- ② A3
- ③ A2
- ④ A1

11. 下図のような実験結果がシートに記録されているとき、条件1と条件2の結果を集計する。実験回数は、0回以上の不定である。シートに記載されているすべての結果を集計するプログラムとして、最もふさわしいものを選択しなさい。ただし、各プログラム片において、「集計処理」は正しく記載されているものとする。

	A	B	C	D
1		条件 1	条件 2	集計結果
2	1回め	106.33	216.07	
3	2回め	110.95	219.48	
4				
5				

- ①
 

```
Do While Not IsEmpty(ActiveCell)
    集計処理
ActiveCell.Offset(1, 0).Activate
Loop
```
- ②
 

```
Do
    集計処理
ActiveCell.Offset(1, 0).Activate
Loop While Not IsEmpty(ActiveCell)
```
- ③
 

```
Do Until Not IsEmpty(ActiveCell)
    集計処理
ActiveCell.Offset(1, 0).Activate
Loop
```

④

**Do**

**集計処理**

```
ActiveCell.Offset(1, 0).Activate  
Loop Until Not IsEmpty(ActiveCell)
```

12. 以下のプログラム片を実行するとき、処理 1 は何回実行されるか。次から選びなさい。

```
For Each One In Range("A1:C4")  
    処理 1  
Next One
```

① 12

② 4

③ 3

④ 1

13. 以下のプログラム片を実行しようとしたところ、実行時エラーとなった。その原因として、最もふさわしいものを選択しなさい。

```
Dim Data As Collection  
Data.Add "A"  
Data.Add 10
```

① 変数 Data がインスタンス化されていないから。

② 異なる型を Add しようとしたから。

③ Add を利用できない型だから。

④ Add を 2 回行おうとしたから。

14. クラス A 内に private メンバ変数 x\_ が宣言されているとき、以下のようなプログラム片が動作するためには、ClassA にはどのような記述が必要か。最もふさわしいものを選択しなさい。

```
Dim A as ClassA  
A = new ClassA  
A.X = 10
```

①

```
Public Property Let X(x1 As Integer)  
    x_ = x1  
End Property
```

②

```
Public Property Let X(x1 As Integer)  
    X = x1  
End Property
```

③

```
Public Property Get X(x1 As Integer)
    X = x1
End Property
```

④

```
Public Property Get X(x1 As Integer)
    x_ = x1
End Property
```

15. ユーザフォーム上に配置されたテキストボックス TextBox1 がフォーカスされたとき、TextBox1 の背景色を変更するためには、どの記述が必要か。最もふさわしいものを選択しなさい。

①

```
Private Sub
    TextBox1_Enter()
```

②

```
Private Sub
    TextBox1_Change()
```

③

```
Private Sub
    TextBox1_DropButtonClicked()
```

④

```
Private Sub
    TextBox1_MouseDown()
```

16. 作成したユーザフォーム UserForm1 を表示するとき、表示中でもシートの操作ができるようにするための記述として、最もふさわしいものを選択しなさい。

① UserForm1.Show 0

② UserForm1.Show vbModal

③ UserForm1.Show

④ UserForm1.Show 1

17. 以下のプログラム片の説明として誤っているものはどれか。選びなさい。

```
Dim Data As Collection
Set Data = New Collection
Data.Add "Co"
Data.Add "St"
Data.Add Item:="2018", Key:="Year"
```

① Data(Year)により 2018 を取り出すことができる

② Data("Year")により 2018 を取り出すことができる

- ③ Data(3)により 2018 を取り出すことができる
- ④ Data.Item("Year")により 2018 を取り出すことができる

18. 次のプログラム片を実行したのち、X の値が 2 になるのは次はどれか。選びなさい。

①

```
Dim X As Integer
X = 1
Foo (X)
```

```
Sub Foo(n As Integer)
    n = n + 1
End Sub
```

②

```
Dim X As Integer
X = 1
Foo X
```

```
Sub Foo(n As Integer)
    n = n + 1
End Sub
```

③

```
Dim X As Integer
X = 1
Foo (X)
```

```
Sub Foo(ByVal n As Integer)
    n = n + 1
End Sub
```

④

```
Dim X As Integer
X = 1
Foo X
```

```
Sub Foo(ByVal n As Integer)
    n = n + 1
End Sub
```

19. 下記のプログラム片を実行した結果として、最もふさわしいものを選択しなさい。

```
Dim i As Integer
For i = -2 To 2
    Cells(i + 3, 1).Value = 10 / i
    On Error GoTo label1
Next i
```

```
Exit Sub
```

label1:

```
Cells(i + 3, 1).Value = "Error"
Resume Next
```

①

	A
1	-5
2	-10
3	Error
4	10
5	5

②

	A
1	-5
2	-10
3	Error
4	
5	

③

	A
1	-5
2	-10
3	10
4	5
5	

④

	A
1	-5
2	-10
3	
4	
5	

20. 下記のプログラム片は、Function Foo の結果を myRange に受け取る。(A)に入れる最もふさわしいものを選択しなさい。

```
Dim myRange As Range
```

```
(A)
```

```
Function Foo() As Range
```

```
    処理
```

```
End Function
```

- ① set MyRange = Foo
- ② MyRange = Foo
- ③ Foo = MyRange
- ④ Foo(MyRange)

【問題1】

河川の水位を上流から下流まで4地点で毎時間同時に計測しており、時刻ごとに4地点のグラフが画像として保存されている。画像のファイル名は時系列に g01.jpg から連番であり、最後は g30.jpg である。画像は Excel ファイルが保存されているフォルダ内の images フォルダに保存されている。

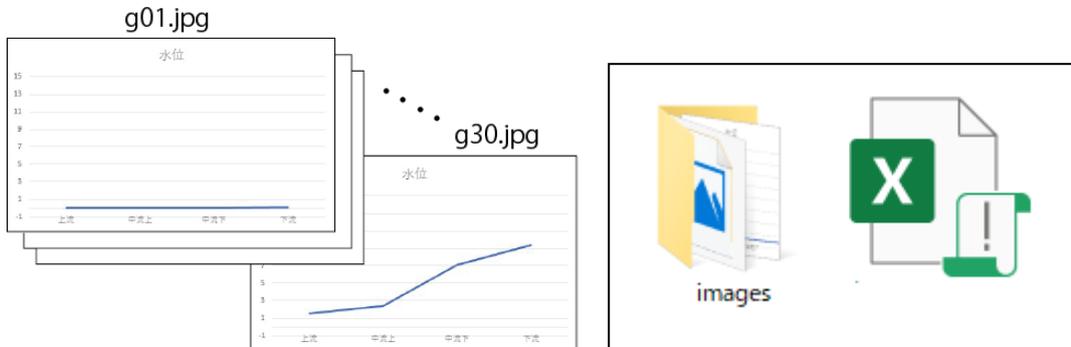


図1 グラフ画像の保存フォルダ

図2に示すユーザフォームを作成し、イメージ領域にグラフを表示、「次へ」ボタンをクリックすると、次の時刻のグラフに切り替える。同様に、「前へ」ボタンをクリックすると、1つ前の時刻のグラフを表示する。最後のグラフが表示されているとき、「次へ」ボタンがクリックされたら、メッセージボックスを表示して、最後の画像であることを知らせる。同様に、最初のグラフが表示されているとき、「前へ」ボタンがクリックされたら、メッセージボックスを表示して、最初の画像であることを知らせる(図3)。

「次へ」ボタン「前へ」ボタンと同様の機能を持つスクロールバーを配置する。スクロールバーが最も左のときは最初のグラフを表示し、最も右のときは最後のグラフを表示する。スクロールバーのハンドル位置は、表示されているグラフを表す位置でなければならない。つまり、「次へ」ボタンと「前へ」ボタンがクリックされたときも、スクロールバーのハンドル位置を変更する。

「終了」ボタンがクリックされたら、画面を閉じる。

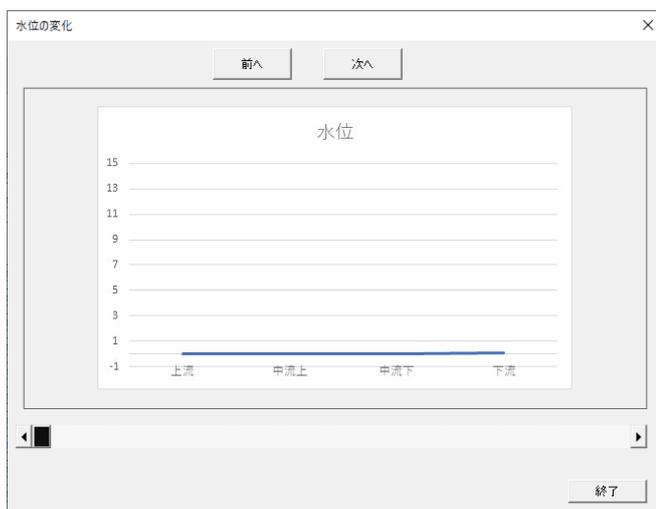


図2 最初のグラフを表示



図 3 メッセージボックスの表示

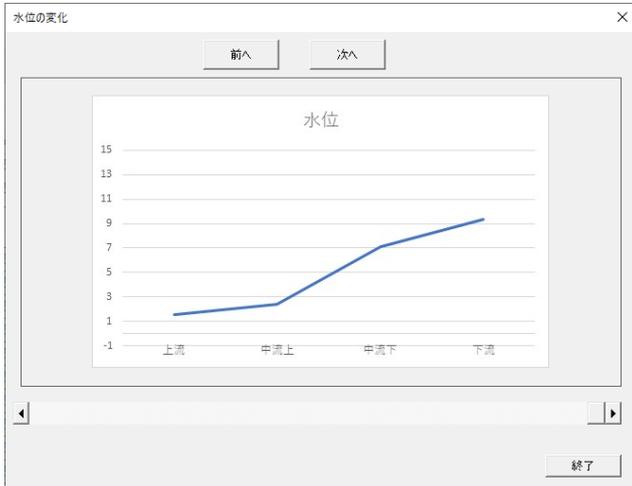


図 4 最後のグラフを表示

入力画面上の各コントロールのオブジェクト名は図5のとおりである。

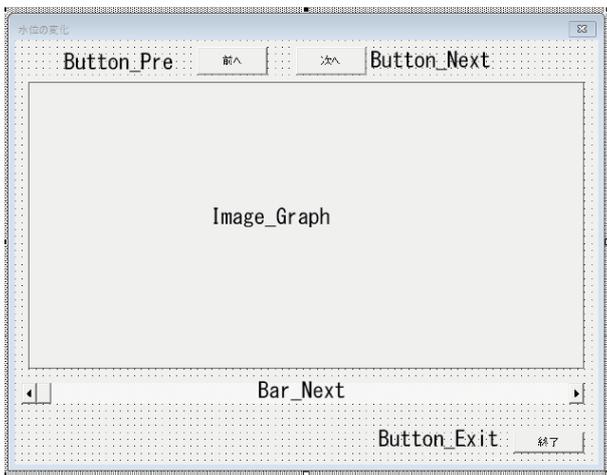


図 5 コントロールのオブジェクト名

プログラムでは、以下のグローバル変数を利用している。

型	変数名	説明
Integer	N	最終のグラフ画像の番号
Integer	I	現在表示しているグラフ画像の番号

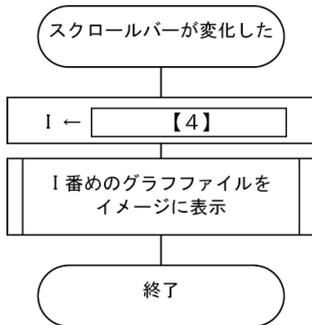
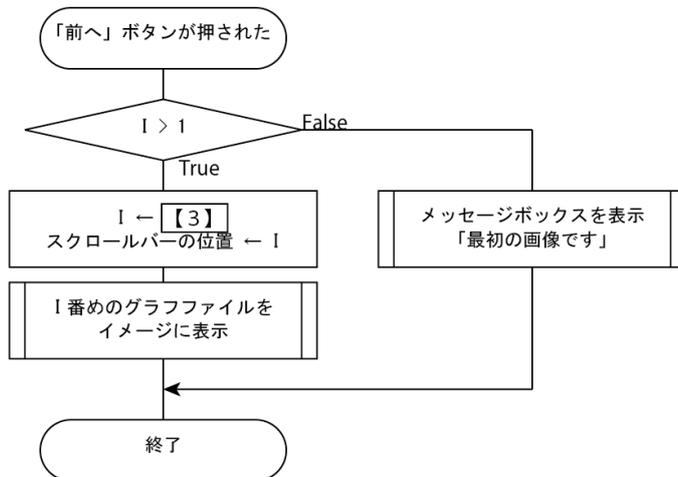
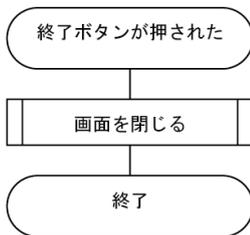
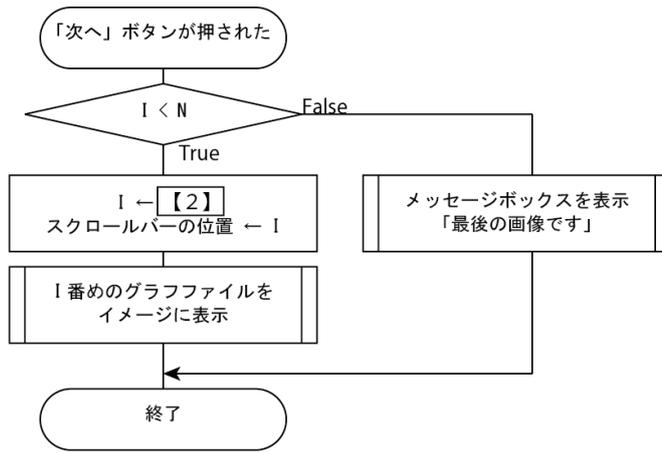
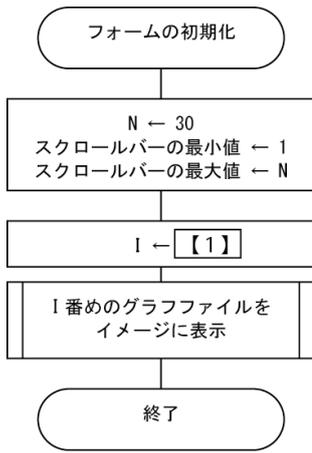
マクロ VBA では、以下のプロパティが利用できる。

型	プロパティ	説明
String	ThisWorkbook.Path	Excel ファイルが保存されているフォルダの絶対パス

マクロ VBA では、以下の関数が利用できる。

型	関数の形式と説明
String	Format(式, “書式指定”) 式を指定の書式の文字列にする。

## 【アルゴリズム】



## 【プログラム】

```

Sub Macro()
    UserForm1.Show
End Sub
  
```

```

Dim N As Integer
Dim I As Integer

Private Sub UserForm_Initialize()
    N = 30
    Bar_Next.Min = 1
    Bar_Next.Max = N
  
```

```

I = 【1】
One_Draw
End Sub

Private Sub Button_Exit_Click()
    Unload UserForm1
End Sub

Private Sub Button_Next_Click()
    If I < N Then
        I = 【2】
        Bar_Next.Value = I
        One_Draw
    Else
        MsgBox "最後の画像です", vbOKOnly, "メッセージ"
    End If
End Sub

Private Sub Button_Pre_Click()
    If I > 1 Then
        I = 【3】
        Bar_Next.Value = I
        One_Draw
    Else
        MsgBox "最初の画像です", vbOKOnly, "メッセージ"
    End If
End Sub

Private Sub Bar_Next_Change()
    I = 【4】
    One_Draw
End Sub

Private Sub One_Draw()
    Dim Path As String
    Dim FileName As String

```

```
FileName = "%g" + Format(l, "00") + ".jpg"  
Path = ThisWorkbook.Path + "%images" + FileName  
Image_Graph.Picture = LoadPicture(Path)
```

```
End Sub
```

次の中から、上の空欄【1】～【4】に入る最も適切なものを選びなさい。なお、プログラムでは、データは配列の初期化により与えられる。

- 【1】 (1) 1 (2) 0  
(3) N (4) 30
- 【2】 (1) l + 1 (2) 1  
(3) l - 1 (4) Bar\_Next.Value
- 【3】 (1) l - 1 (2) 1  
(3) l + 1 (4) Bar\_Next.Value
- 【4】 (1) Bar\_Next.Value (2) l + 1  
(3) l - 1 (4) 30

【問題 2】

列車の座席を自動販売機で予約することをシミュレートするプログラムである。予約のルールは以下のとおりである。

- (1) 停車駅と座席の座席名は、図 1 のシートから与えられるものとする。
- (2) 図 2 の予約画面にて、乗車駅番号と降車駅番号を入力し、検索ボタンをクリックすると、購入可能な座席番号を表示する。
- (3) 座席番号を入力して予約ボタンをクリックすると、予約が完了する。予約可能な座席がないときは、予約ボタンは無効となりクリックすることはできない。
- (4) 乗車駅と降車駅の間が連続して空席であれば予約することができる。
- (5) 切符は 1 回の操作で 1 枚しか購入することはできない。

なお、問題簡略化のため、車両の情報は省略しており、また、入力に誤りはないものとする。

	A	B	C	D	E	F
1						
2	駅番号	駅名	1A	2A	3A	4A
3		1 上野				
4		2 柏				
5		3 土浦				
6		4 石岡				
7		5 友部				
8						

図 6 駅名・座席名指定の例（ときわ 55 号）

図 7 予約操作画面

例えば、座席番号 4A に上野から土浦まで予約が確定しているとき、土浦から友部までは新たに予約をすることができるが、柏から友部までの予約はできない。

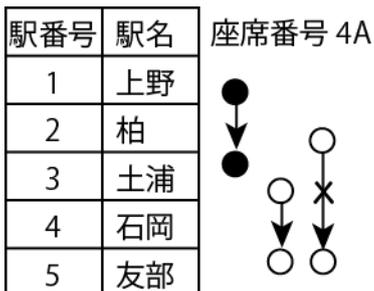


図 8 予約例

画面に配置されたコントロールの名称は図4のとおりである。

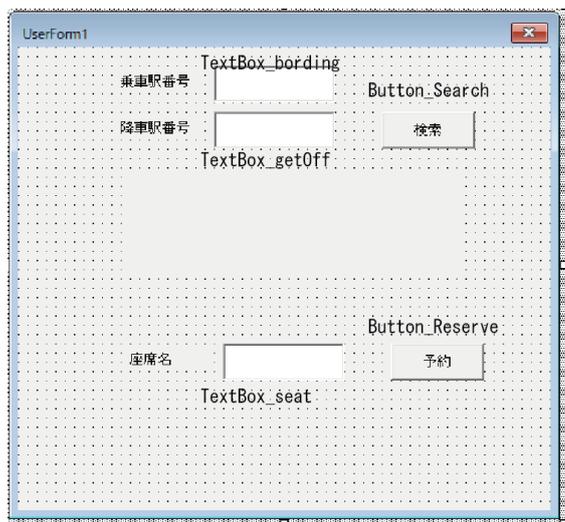


図 9 画面上のコントロールの名称

このプログラムは、以下のクラスから構成されている。

表 1 クラス Seat

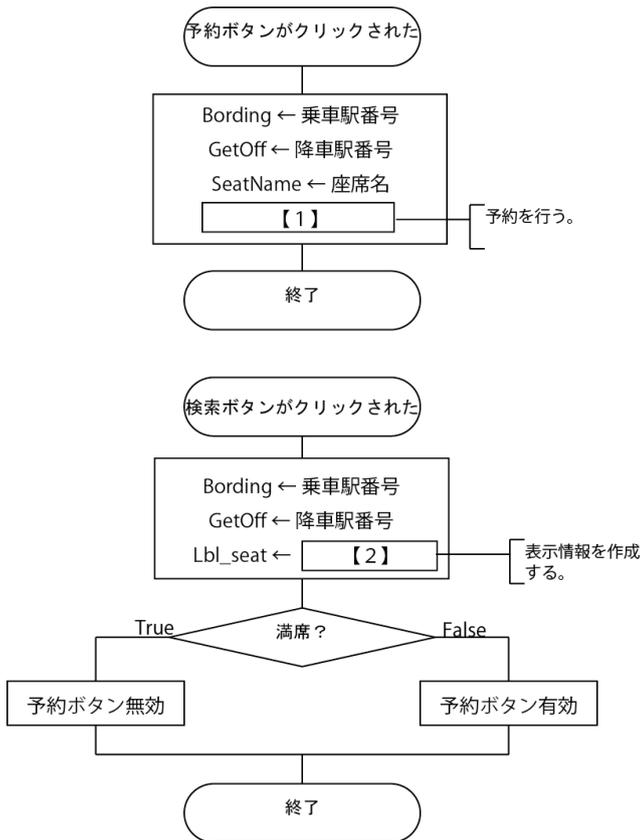
フィールド			
修飾子	型	名称	説明
Private	Object	station_	各座席について、駅間の予約状況を保持する連想配列(Dictionary)である。駅番号を添字とする。空席の時は false、予約済みの時は true とする。列車が停車する駅の数- 1 個の要素を持つ。
Private	Integer	col_	シート上の列番号を保持する。
コンストラクタ			
修飾子	形式と説明		
Public	station_ を初期化する。初期値は全要素 False にする。		
プロパティ			
修飾子	形式と説明		
Public	Let	Col(ByVal i As Integer)	シート上の列番号を与える
メソッド			
修飾子	型	形式と説明	
Public		Init() 該当するシートの初期化を行う	
Public	Boolean	Function isEmpty(Bording As Integer, GetOff As Integer) As Boolean bording : 乗車駅番号 getOff : 降車駅番号 指定の乗車区間の予約が可能かどうかを調べる。乗車区間が空席で予約が可能であれば True を返す。1 区間でも予約済みであれば False を返す。	
public		Reserved(Bording As Integer, GetOff As Integer)	

		boarding : 乗車駅番号 getOff : 降車駅番号 予約区間の stations_ を予約済みである True に変更する。同時にシート上の該当するセルを「予約済」とする。
--	--	--

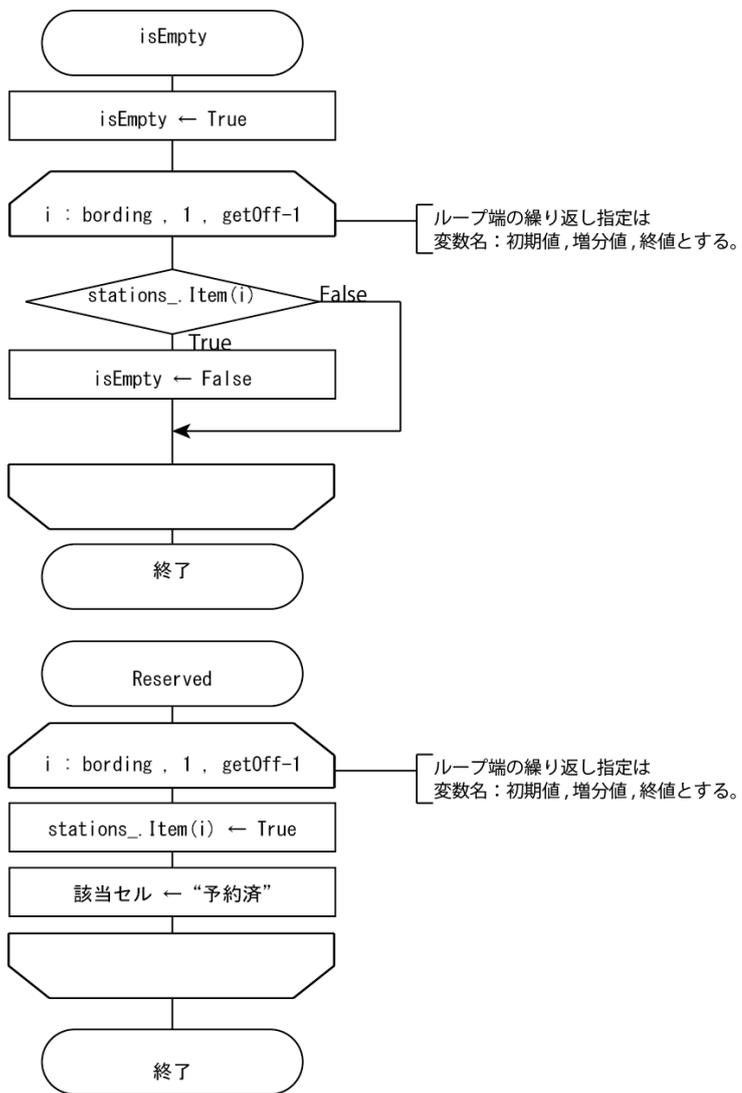
表 2 クラス Train

フィールド			
修飾子	型	名称	説明
Private	Object	seats_	座席名を添字とする連想配列である。要素として Seat クラスを持つ。
Private	Boolean	full_	乗車区間が指定されたとき、その区間で満席であれば False を、1 席以上の空席があれば True とする。
コンストラクタ			
修飾子	形式と説明		
Public	seats_ を初期化する。seats_ は、座席名を添字とする連想配列を形成し、要素として Seat クラスを生成する。		
プロパティ			
修飾子		形式と説明	
Public	Get	SeatsItem(Name As String) As Seat seats_ 連想配列より、指定の要素を 1 つ取り出す。	
Public	Get	Full() As Boolean full_ を取り出す。True のときは、満席であり、False のときは予約可能である。	
メソッド			
修飾子	型	形式と説明	
Public	String	GetState(Bording As Integer, GetOff As Integer) 予約画面に表示する文字列を形成する。座席名とともに、各座席について指定区間が空席であれば○、1 区間で予約済であれば、× が文字列内に埋め込まれる。	
Public	void	Reserved(Bording As Integer, GetOff As Integer, SeatName As String) boarding : 乗車駅番号 getOff : 降車駅番号 SeatName : 座席名 該当する Seat クラスのオブジェクトに対して、予約を行う。	

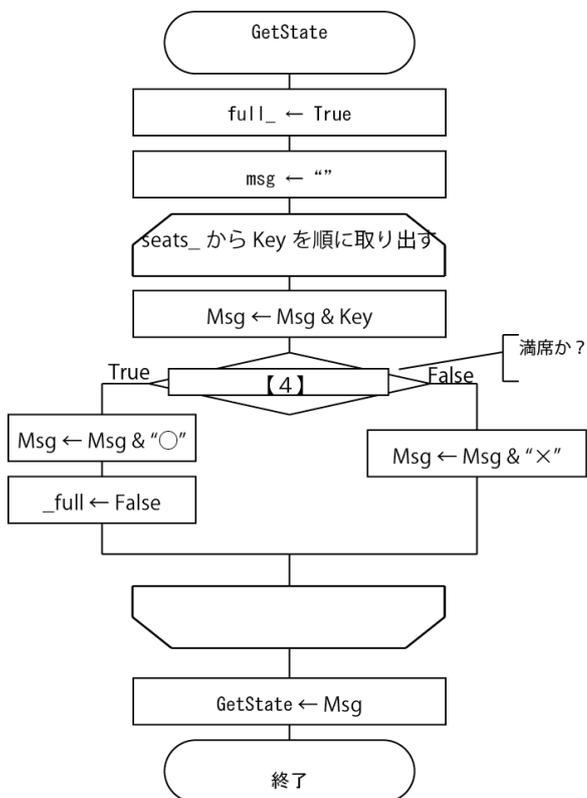
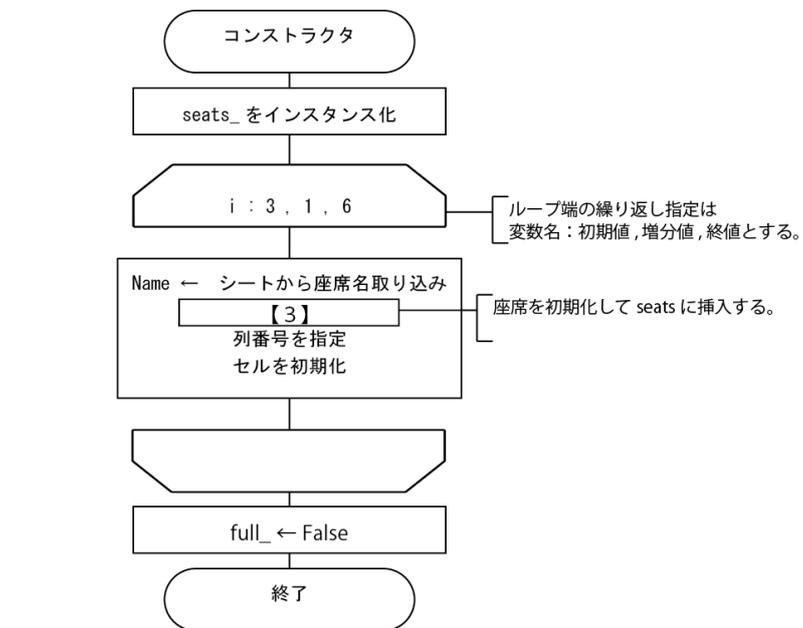
## 【アルゴリズム】



SEAT クラス



Train クラス



## 【プログラム】

### UserForm1 のコード

```
Dim Tokiwa55 As Train

Private Sub UserForm_Initialize()
    Set Tokiwa55 = New Train
End Sub

Private Sub Button_reserve_Click()
    Dim Bording As Integer
    Dim GetOff As Integer
    Dim SeatName As String

    Bording = TextBox_bording.Text
    GetOff = TextBox_getOff.Text
    SeatName = TextBox_seat.Text
    

End Sub

Private Sub Button_Search_Click()
    Dim Bording As Integer
    Dim GetOff As Integer

    Bording = TextBox_bording.Text
    GetOff = TextBox_getOff.Text

    Lbl_seat.Caption = Tokiwa55.GetState(Bording, GetOff)
    If  Then
        Button_reserve.Enabled = False
    Else
        Button_reserve.Enabled = True
    End If
End Sub
```

### Seat クラス

```
Private col_ As Integer
Private stations_ As Object

Private Sub class_initialize()
    Dim i As Integer
    Set stations_ = CreateObject("Scripting.Dictionary")
    For i = 1 To 4
        stations_.Add Key:=i, Item:=False
    Next i
End Sub
```

```

Public Property Let Col(ByVal i As Integer)
    col_ = i
End Property

Public Sub Init()
    Dim i As Integer
    For i = 3 To 6
        Cells(i, col_).Value = ""
    Next i
End Sub

Public Function isEmpty(Bording As Integer, GetOff As Integer) As Boolean
    Dim i As Integer
    isEmpty = True
    For i = Bording To GetOff - 1
        If stations_.Item(i) Then
            isEmpty = False
        End If
    Next i
End Function

Public Sub Reserved(Bording As Integer, GetOff As Integer)
    Dim i As Integer
    For i = Bording To GetOff - 1
        stations_.Item(i) = True
        Cells(i + 2, col_).Value = "予約済"
    Next i
End Sub

```

## Train クラス

```

Private seats_ As Object
Private full_ As Boolean

Private Sub class_initialize()
    Dim i As Integer
    Dim Name As String

    Set seats_ = CreateObject("Scripting.Dictionary")
    For i = 3 To 6

```

```
Name = Cells(2, i).Value
```

```
    【 3 】
```

```
    seats_.Item(Name).Col = i
```

```
    seats_.Item(Name).Init
```

```
Next i
```

```
full_ = False
```

```
End Sub
```

```
Public Property Get SeatsItem(Name As String) As Seat
```

```
    Seats = seats_.Item(Name)
```

```
End Property
```

```
Public Property Get Full() As Boolean
```

```
    Full = full_
```

```
End Property
```

```
Public Function GetState(Bording As Integer, GetOff As Integer) As String
```

```
    Dim Msg As String
```

```
    full_ = True
```

```
    Msg = ""
```

```
    For Each Key In seats_
```

```
        Msg = Msg & Key & " : "
```

```
        If 【 4 】 Then
```

```
            Msg = Msg + "○ "
```

```
            full_ = False
```

```
        Else
```

```
            Msg = Msg + "× "
```

```
        End If
```

```
    Next Key
```

```
    GetState = Msg
```

```
End Function
```

```
Public Sub Reserved(Bording As Integer, GetOff As Integer, SeatName As String)
```

```
    Dim i As Integer
```

```
    seats_.Item(SeatName).Reserved Bording, GetOff
```

次の中から、上の空欄【1】～【4】に入る最も適切なものを選びなさい。

- 【1】 (1) Tokiwa55.Reserved Bording, GetOff, SeatName  
(2) Train.Reserved Bording, GetOff, SeatName  
(3) Seat.Reserved Bording, GetOff, SeatName  
(4) Reserved Bording, GetOff, SeatName

- 【2】 (1) Tokiwa55.Full  
(2) Train.Full  
(3) Full  
(4) full\_

- 【3】 (1) seats\_.Add Key:=Name, Item:=New Seat  
(2) seats\_.Add New Seat  
(3) seats\_.Add Name  
(4) seats\_.Item(Name)=New Seat

- 【4】 (1) seats\_.Item(Key).isEmpty(Bording, GetOff)  
(2) seats\_(Key).isEmpty(Bording, GetOff)  
(3) Seats(Key).isEmpty(Bording, GetOff)  
(4) Seats.Item(Key).isEmpty(Bording, GetOff)

第3問（読解：1問）

【問題】

社会現象をシミュレートするときに欠かすことができないのが乱数である。マクロ VBA は一様乱数を発生する関数が整備されており、これを使って各種乱数を発生することができる。一様乱数を発生する関数は以下のとおりである。

戻り値	関数の形式	機能
Single 型	Rnd	0 以上 1 未満の 1 つの浮動小数点数を得る。

以下は乱数を 1 つ発生するマクロである。

【プログラム】

```
Function Macro(a As Integer, b As Single) As Integer
    Dim i As Integer
    Dim k As Integer

    k = 0
    For i = 1 To a
        k = k + One(b)
    Next i
    Macro = k
End Function

Function One(b As Single) As Integer
    If Rnd < b Then
        One = 1
    Else
        One = 0
    End If
End Function
```

上のプログラムにより生成される乱数の特徴を表しているのは次のうちどれか。最もふさわしいものを選択しなさい。

【選択肢】

- 【1】 Macro を何度も呼び出すことで、成功する確率が  $b$  である事象を  $a$  回行ったとき、成功する回数の分布が得られる。
- 【2】 Macro を何度も呼び出しことで、時間  $b$  のあいだに  $a$  回事象が発生する分布が得られる。
- 【3】 Macro を何度も呼び出すことで、平均  $a$ 、分散  $b$  の正規分布が得られる。
- 【4】 Macro を何度も呼び出すことで、生起率  $b$  で事象が起きるとき、次に起きるまでの時間の分布が得られる。

# 実技科目

## 【問題 1】

スタート地点からゴール地点に移動するにあたり、複数のルートの中から、もっとも短時間で移動が見込まれるルートを選択する。ただし、途中で休憩できる施設があるルートでなければならない。例えば、下表のように5つのルートがある場合、ルートCを選択する。

表 3 ルート一覧の例

	ルート A	ルート B	ルート C	ルート D	ルート E
休憩施設	なし	あり	あり	なし	あり
見込み時間	3.8 時間	4.5 時間	4.2 時間	3.7 時間	4.4 時間

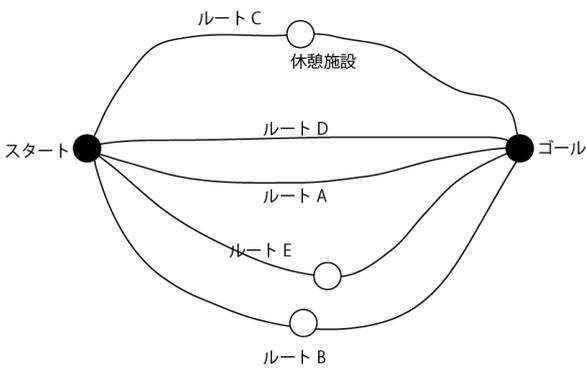


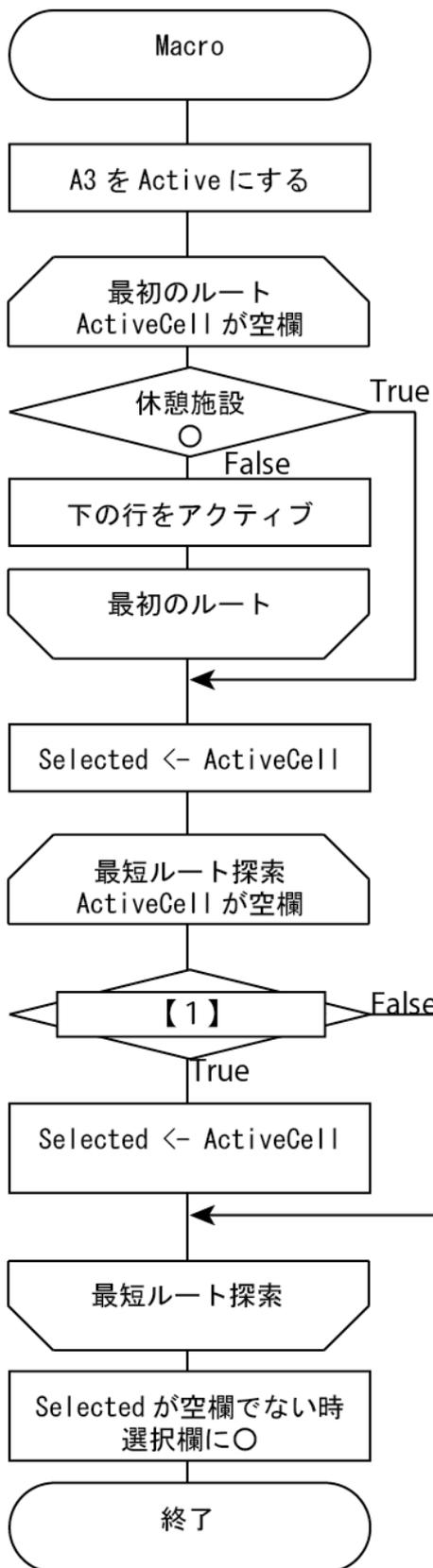
図 10 ルート例

各地点に関する情報は、図 2 のようにシートに記録されている。

	A	B	C	D
1				
2	名称	休憩施設	見込み時間 (時間)	選択
3	ルート A	×	3.8	
4	ルート B	○	4.5	
5	ルート C	○	4.2	
6	ルート D	×	3.7	
7	ルート E	○	4.4	
8				

図 11 ルート例の記録

## 【流れ図】



..... [ 休憩施設があり、Selected が示す見込み時間より短時間のルートするとき Selected を書き換える。 ]

以下のプログラムの空欄に入る適切なプログラム片を入力してプログラムを完成させなさい。

```

Sub Macro()
  Dim Selected As Range

```

```
Range("A3").Activate
Do While Not IsEmpty(ActiveCell)
    If ActiveCell.Offset(0, 1).Value = "○" Then
        Exit Do
    End If
    ActiveCell.Offset(1, 0).Activate
Loop

Set Selected = ActiveCell
Do While Not IsEmpty(ActiveCell)
    If ActiveCell.Offset(0, 1).Value = "○" And  Then
        Set Selected = ActiveCell
    End If
    ActiveCell.Offset(1, 0).Activate
Loop

If Not IsEmpty(Selected) Then
    Selected.Offset(0, 3).Value = "○"
End If
End Sub
```

【問題 2】

図 1 のような迷路をスタート地点からゴール地点に最短で移動する経路をすべて列挙するプログラムである。  
■は壁であり、進むことができない。また、最短で移動するために、右と下にのみ移動できる。

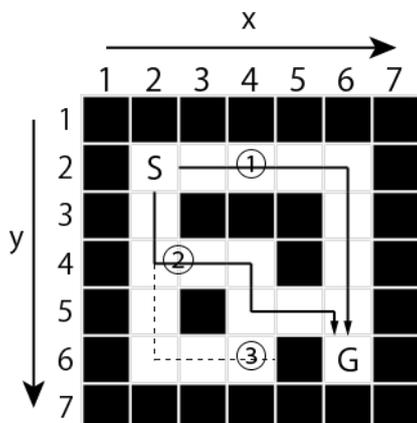


図 12 迷路の例

図 1 の例では、①と②はゴールにたどり着くことができるが、③は、途中で行き止まりになってしまい、ゴールにたどり着くことができない。

プログラムでは、壁の構成は、図 2 のようにシートに与えられる。「1」は壁であり、「0」は通路である。スタート地点 S は、(2,2)、ゴール地点 G は(6,6)である。

	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	1
3	1	0	1	1	1	0	1
4	1	0	0	0	1	0	1
5	1	0	1	0	0	0	1
6	1	0	0	0	1	0	1
7	1	1	1	1	1	1	1

図 13 シート上の壁定義

地点を表す x と y の組はクラス Position を利用する。Position の仕様は表 1 のとおりである。  
通過した位置は、Position クラスの配列 route に記録していく。ゴールに到達したときは、配列 route の内容を先頭から順にシート上に出力する。その後、順に戻りながら、次のルートを探す。このとき、route の内容は 1 つずつ破棄する。行き止まりに阻まれたときも、同様に、route の内容を破棄しながら戻っていく。なお、配列 route は十分な領域が確保されており、領域チェックは省略している。

表 4 Position クラス

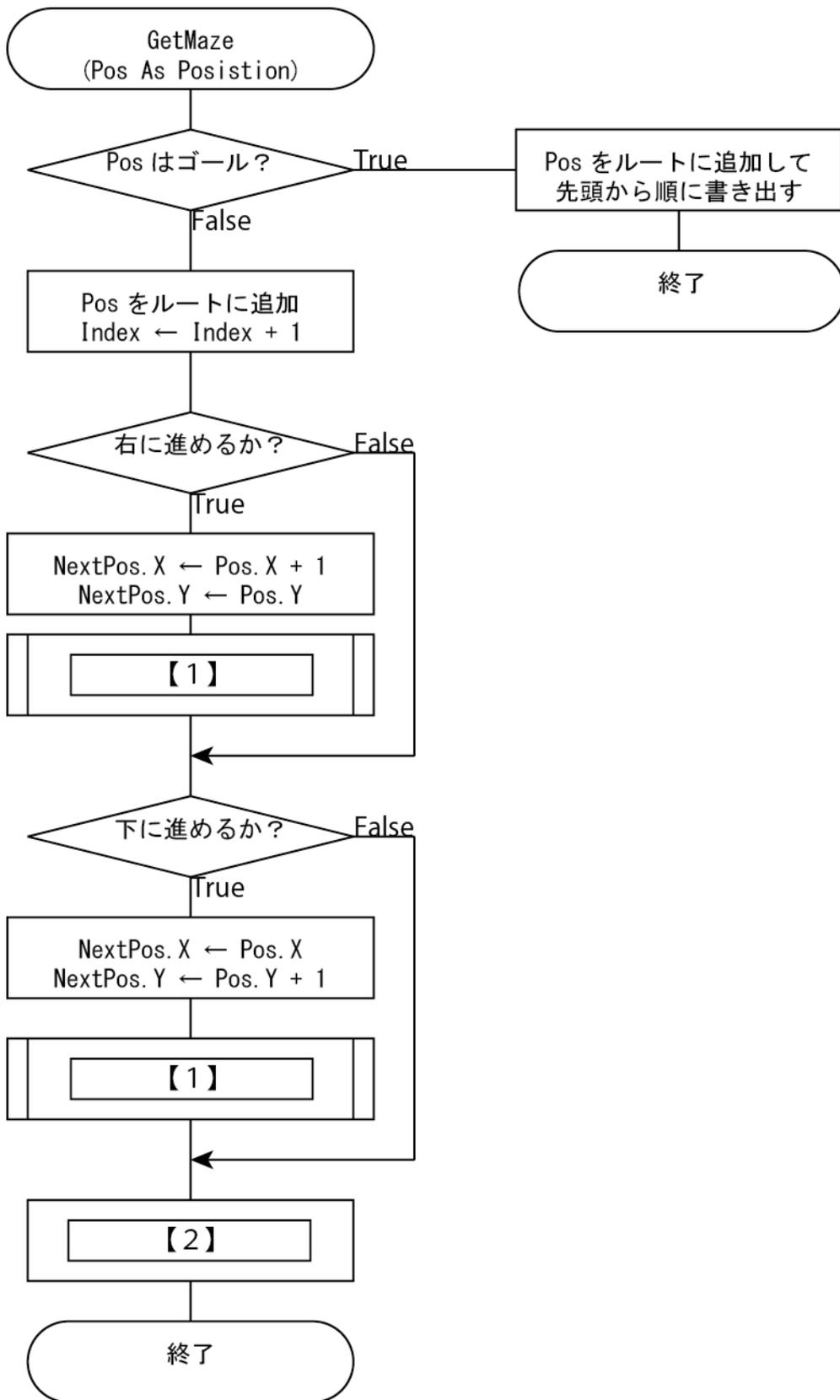
プロパティ			
修飾子	型	名称	説明
Public	Integer	X	横方向の添字
Public	Integer	Y	縦方向の添字
メソッド			
修飾子	戻り値	メソッドと説明	
Public	Boolean	IsGoal() ゴール位置であれば true そうでなければ false を返す。	

プログラムで定義し、利用しているサブルーチンの仕様は以下のとおりである。

表 5 定義しているサブルーチン

戻り値	関数の形式と機能
	GetMaze(Pos As Position) スタート位置からゴール位置までの経路を表示する。 Pos : 開始位置を指定する。
	OutputRoute(Index As Integer) 経路が格納されている配列 route の内容を Index の前までシートに出力する。
Boolean	CheckGo(Pos As Position, Direction As Integer) pos の右または下の壁の状態から進めるかどうかを返す。 direction 1 : 右 2 : 下

【流れ図】



以下のプログラムの空欄に入る適切なプログラム片を入力してプログラムを完成させなさい。

```
Dim Route(30) As Position
Dim Index As Integer

Sub Macro()
    Dim Pos As Position

    Range("I1").Activate
    Index = 0
    Set Pos = New Position
    Pos.X = 2
    Pos.Y = 2
    GetMaze Pos
End Sub

Sub GetMaze(Pos As Position)
    Dim NextPos As Position

    If Pos.IsGoal Then
        Set Route(Index) = Pos
        OutputRoute (Index + 1)
        Exit Sub
    End If

    Set Route(Index) = Pos
    Index = Index + 1

    If CheckGo(Pos, 1) Then
        Set NextPos = New Position
        NextPos.X = Pos.X + 1
        NextPos.Y = Pos.Y

|       |
|-------|
| 【 1 】 |
|-------|



    End If

    If CheckGo(Pos, 2) Then
        Set NextPos = New Position
        NextPos.X = Pos.X
        NextPos.Y = Pos.Y + 1
```

【 1 】

End If

【 2 】

End Sub

Sub OutputRoute(Index As Integer)

ActiveCell.Offset(0, 0).Value = "y"

ActiveCell.Offset(0, 1).Value = "x"

For i = 0 To Index - 1

ActiveCell.Offset(i + 1, 0).Value = Route(i).Y

ActiveCell.Offset(i + 1, 1).Value = Route(i).X

Next i

ActiveCell.Offset(0, 3).Activate

End Sub

Function CheckGo(Pos As Position, Direction As Integer) As Boolean

Dim i As Integer

Dim j As Integer

Select Case Direction

Case 1

i = Pos.X + 1

j = Pos.Y

Case 2

i = Pos.X

j = Pos.Y + 1

Case Else

i = Pos.X

j = Pos.Y

End Select

If Cells(j, i).Value = 1 Then

CheckGo = False

Else

CheckGo = True

End If

End Function

Position クラス

```
Public X As Integer
Public Y As Integer

Public Function IsGoal() As Boolean
    If X = 6 And Y = 6 Then
        IsGoal = True
    Else
        IsGoal = False
    End If
End Function
```

### 【問題3】

シャーレの中でバクテリアがどのように増殖するのかをシミュレートするプログラムである。シャーレを微小な空間に分け、その小さな一つひとつの空間にバクテリアが最大1匹存在できるものとする。図1では初期状態として、6匹のバクテリアが存在している。

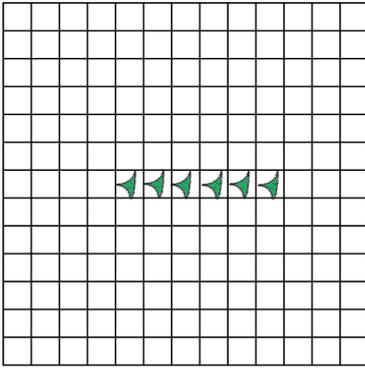


図 14 微小空間にいるバクテリア

バクテリアは、環境により次のように変化する。

- ① バクテリアは、図2に示す近傍8個のセルのうち2個または3個のセルにバクテリアが存在すれば、生き続けられる。しかし、2個未満のときは、寂しくて死んでしまう。また、4個以上の時は、過密すぎて死んでしまう。
- ② バクテリアが存在しないセルでは、図3に示す近傍8個のセルのうち3個のセルにバクテリアが要れば、分裂により新たに誕生することができる。

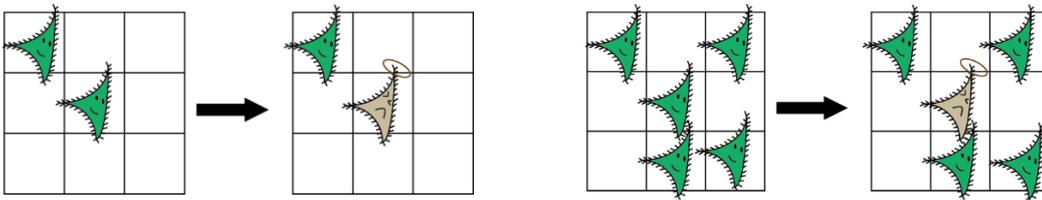


図 15 周りのバクテリアが少なくても多くても死んでしまう

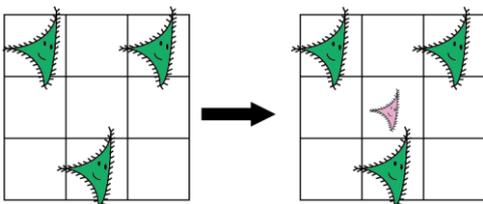


図 16 近傍が3匹であれば赤ちゃん誕生

以上のルールに従って、時間が進んだとき、バクテリアがどのように変化しているかをシミュレートする。ただし、領域の外側にはバクテリアは生存せず、生誕することもできないものとする。

プログラムでは、13×13の領域に分けており、定数 TATE と YOKO で定義している。バクテリアの初期状態は、Excel シート上で与える。黒く塗りつぶしたセルにバクテリアが生存していることを示している。この様子を図4に示す。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													

図 17 Excel シートに与えられた初期状態

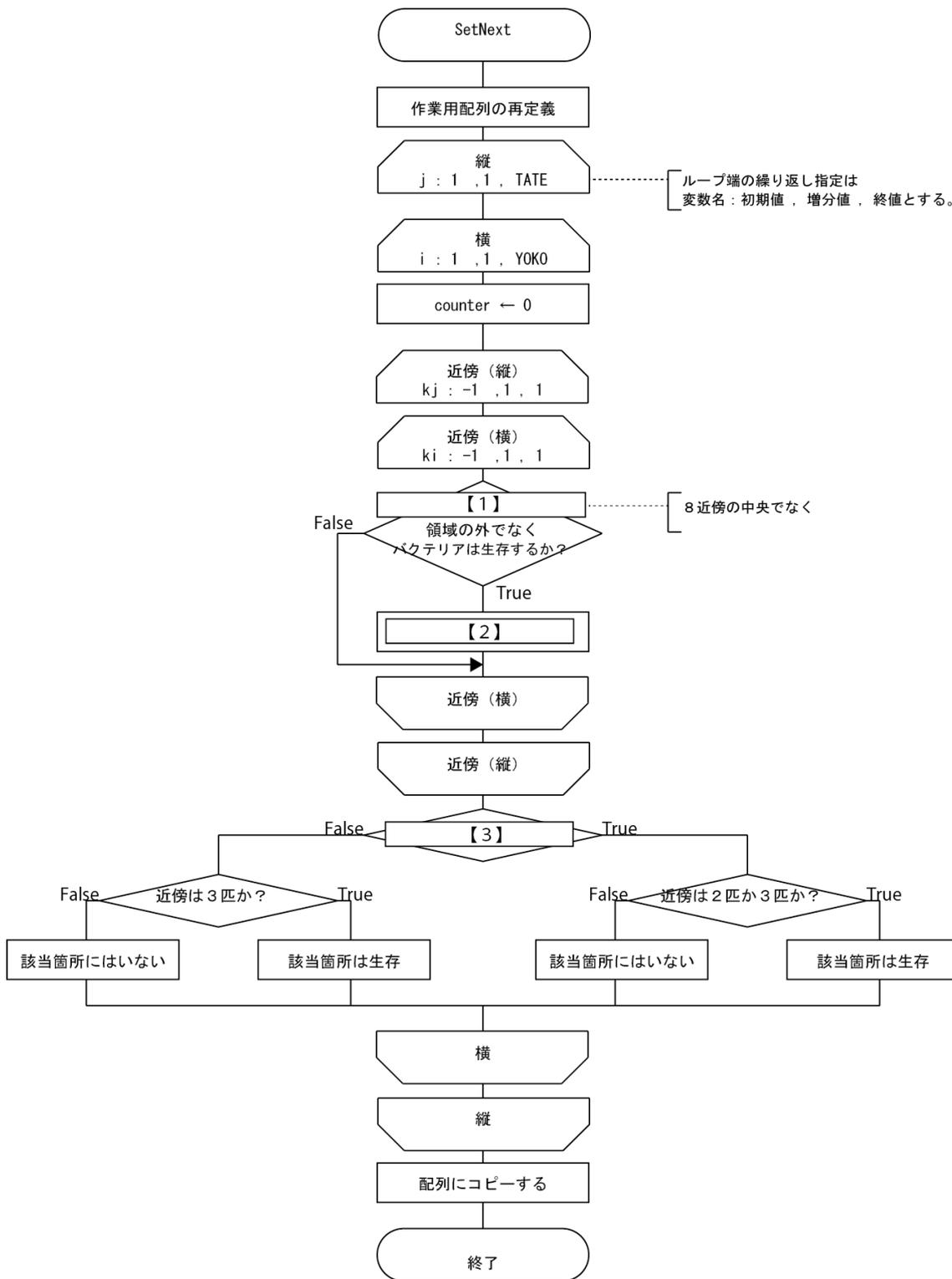
微小空間の管理はクラス World で行う。World 内の 2 次元配列 UniCell に各微小空間のバクテリアの状態を記録しており、バクテリアが生きている状態を true、いない状態を false とする。

バクテリアの状態を 1 2 回更新したときのバクテリアの状態を Excel シート上に表示する。World クラスの様子は表 1 のとおりである。

表 6 World クラス

フィールド			
修飾子	型	名称	説明
private	int	TATE	縦方向の要素数
private	int	YOKO	横方向の要素数
private	Boolean[][]	UniCell	バクテリアの状態を記録する配列
コンストラクタと説明			
Private Sub class_initialize() TATE, YOKO を指定する。			
メソッド			
修飾子	型	メソッドと説明	
Public	void	GetCell() Excel シートから初期状態を配列 UniCell に取り込む。	
Public	void	SetCell() UniCell の状態を Excel シート上に表にする。	
Public	void	setNext() バクテリアの状態を 1 回更新する。配列 cell が更新される。	

【流れ図】



以下のプログラムの空欄に入る適切なプログラム片を入力してプログラムを完成させなさい。

```

Private TATE As Integer
Private YOKO As Integer
Private UniCell() As Boolean

Private Sub class_initialize()
    TATE = 13
    YOKO = 13
  
```

```
End Sub
```

```
Public Sub GetCell()
```

```
    ReDim UniCell(TATE, YOKO)
```

```
    Dim i As Integer
```

```
    Dim j As Integer
```

```
    For j = 1 To TATE
```

```
        For i = 1 To YOKO
```

```
            If Cells(j, i).Interior.Color = vbBlack Then
```

```
                UniCell(j, i) = True
```

```
            Else
```

```
                UniCell(j, i) = False
```

```
            End If
```

```
        Next i
```

```
    Next j
```

```
End Sub
```

```
Public Sub SetCell()
```

```
    Dim i As Integer
```

```
    Dim j As Integer
```

```
    For j = 1 To TATE
```

```
        For i = 1 To YOKO
```

```
            If UniCell(j, i) Then
```

```
                Cells(j, i).Interior.Color = vbBlack
```

```
            Else
```

```
                Cells(j, i).Interior.Color = vbWhite
```

```
            End If
```

```
        Next i
```

```
    Next j
```

```
End Sub
```

```
Public Sub SetNext()
```

```
    Dim NextCell() As Boolean
```

```
    Dim i As Integer
```

```
    Dim j As Integer
```

```
    Dim ki As Integer
```

```
    Dim kj As Integer
```

```
    Dim Counter As Integer
```

```
    ReDim NextCell(TATE, YOKO)
```

```
    For j = 1 To TATE
```

```
        For i = 1 To YOKO
```

```
            Counter = 0
```

```
            For kj = -1 To 1
```

```
                For ki = -1 To 1
```

```
                    If  Then
```

```

                If j + kj > 0 And j + kj <= TATE And i + ki > 0 And_
                                                                i + ki <= YOKO Then
                    If UniCell(j + kj, i + ki) Then
                        【 2 】
                    End If
                End If
            End If
        Next ki
    Next kj

    If 【 3 】 Then
        If Counter = 2 Or Counter = 3 Then
            NextCell(j, i) = True
        Else
            NextCell(j, i) = False
        End If
    Else
        If Counter = 3 Then
            NextCell(j, i) = True
        Else
            NextCell(j, i) = False
        End If
    End If
Next i
Next j

For j = 1 To TATE
    For i = 1 To YOKO
        UniCell(j, i) = NextCell(j, i)
    Next i
Next j
End Sub

```