プログラミング基礎

第6回 ファイル、文字列処理

今日の内容

- ファイル
  - 保存
  - 読み込み
  - 選択ダイアログ
- 文字列処理

ファイルに保存

保存先は「{デスクトップ}/保存.txt」

「{保存先}に保存します」と表示
「こんにちは」を保存先に保存
「さようなら」を保存先に追加保存

- 「デスクトップ」を他の場所に変えてみよう
- 「保存」と「追加保存」の違いを試してみよう

ファイルに一行ずつ保存

出力先ファイルは「{デスクトップ}/保存２.txt」
出力先初期化

番号で１から１０まで繰り返す
　　番号を一行出力

- 「出力先初期化」がない場合も試してみよう

ファイルから読み込む

読込元は「{デスクトップ}/保存.txt」

「{読込元}から読み込みます」と表示
内容に読込元を開く
内容を表示

ファイルを一行ずつ読み込む

ファイルに「{デスクトップ}/保存２.txt」を毎行読む

ファイルを反復
　　対象を表示

ファイル選択ダイアログ

- 「開く」の場合

```
Ｆは「txt」のファイル選択
中身にＦを開く
「{Ｆの長いファイル名取得}の中身です」を表示
中身を表示
```

- 「毎行読む」の場合

```
ファイルは「nako」のファイル選択を毎行読む
ファイルを反復
```

保存ファイル選択ダイアログ

- 「保存」の場合

保存先は「txt」の保存ファイル選択
「こんにちは」を保存先に保存

- 「一行出力」の場合

出力先ファイルは「txt」の保存ファイル選択

練習（１）

- 問１：回答をファイルから読み込むようにする
  - 第４回課題：アンケートの集計
  - 先に「回答.txt」を作っておきましょう
    （１行に１回答）
- 問２：結果をファイルに出力するようにする
  - 第５回課題：なべあつ
  - 「ナベアツ.txt」に出力しましょう

解答例:問1

回答は「{デスクトップ}/回答.txt」を毎行読む

希望者とはハッシュ
回答を反復
　　希望者@対象＝希望者@対象＋１
希望者を表示

- 実は「毎行読む」を「開く」に変えても動作する

解答例:問2

●割り切れる（AがBで）
　　//省略
●付く（AにBが）
　　//省略
●ナベアツ判定（数の）
　　//省略

出力先ファイルは「{デスクトップ}/ナベアツ.txt」
数で１から１００まで繰り返す
　　もし数のナベアツ判定が真ならば「アホ」を一行出力
　　違えば数を一行出力

文字列をつなげる

文字列１は「こんにちは」
文字列２は「さようなら」

「{文字列１}{文字列２}」を表示
文字列１＆文字列２を表示

文字列１に文字列２を追加
文字列１を表示

文字列を整形する

数は123456

数を通貨形式で表示
桁数は１０
数を桁数でゼロ埋めして表示
数を桁数で文字列センタリングして表示
数を桁数で文字列右寄せして表示

練習（2）

- 行番号を付けるプログラムを作ろう
  - 「〇〇.nako」のファイルをダイアログで選択
  - 保存先もダイアログで選択
  - 各行の先頭に「行番号.」を付ける
    （行番号は右が揃うようにする）

解答例

読込元は「nako」のファイル選択を毎行読む
出力先ファイルは「txt」の保存ファイル選択
出力先初期化

行数は０
桁数は３
読込元を反復
　　行数＝行数＋１
　　「{行数を桁数で文字列右寄せ}. {対象}」を一行出力

文字列の中身を取り出す

文は「たけやぶやけた」

「{文}の文字数は{文の文字数}」を表示
文の０から３文字抜き出して表示
文の４文字左部分を表示
文の５文字右部分を表示

数は０
文の文字列分解を反復
　　数＝数＋１
　　「{数}文字目：{対象}」を表示

課題

- 回文かどうか判定するプログラムを作ろう
  - 「回文」は前から読んでも後ろから読んでも同じ文になる文のことです
    - 例：たけやぶやけた（竹やぶ焼けた）
  - 文章はユーザに尋ねるようにします（ひらがなで入力します）

とても余力がある人向け

- 行番号をローマ数字にしてみましょう
  - I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII, XIII, XIV…
  - ローマ字表記（数の）という命令を作る
- 回文の作成に役立つプログラムを考えてみましょう