

テキストエディタを作ろう (実装編：Ruby)

Go Suzuki

目次

1. ファイルを読み書きしよう 楽	1
1.1. 作るモノ	2
1.2. 読み書きする関数を用意しよう	3
1.2.1. 易 <code>openFile</code> のヒント	3
1.2.2. 易 <code>writeFile</code> のヒント	3
1.3. 各処理を作っていこう	3
1.3.1. 易 <code>printLines</code> のヒント	3
1.3.2. 難 <code>appendLines</code> のヒント	3
1.3.3. 易 <code>removeLines</code> のヒント	4
1.4. コマンドを解析しよう	4
1.4.1. 易 ヒント	4
1.5. <code>main</code> を用意しよう	4

ライセンス

この文書は CC-BY である。また、この文書により生じる一切の請求、損害、その他の義務について何らの責任も負わない。

1. ファイルを読み書きしよう 楽

では、ラインエディタと呼ばれるものを最初に作ってみよう。ラインエディタは ed が有名である。

1.1. 作るモノ

```
$ hoge.rb hoge.txt
> p0,3
hoge
piyo
fuga
funya
> a1
puru
pura
.
> p0,5
hoge
piyo
puru
pura
fuga
funya
> r4,100000
> p0,1000
hoge
piyo
puru
> q(プログラム終了)
```

作るラインエディタは、引数に読み書きするファイル名を指定する。> を表示してコマンドを待機する。コマンドは次の通りである。

- (1) q : プログラムを終了する
- (2) w : 変更を保存する (ファイルに書き出す)
- (3) p : 指定された範囲の行を表示する
- (4) r : 指定された範囲の行を削除する
- (5) a : 指定された行の後に入力された行を追加する (. で入力終了.)

範囲指定は、<start>,<end> という書式で行う。<start> 以上 <end> 未満であることを

表す。<start> とだけ渡された場合は、<start> 以上 <start> + 1 未満であると解釈する。

1.2. 読み書きする関数を用意しよう

まずはファイルを読み書きする関数を用意してみよう。

```
openFile(str)
    # ファイル読み込みする。返り値はファイルの内容
writeFile(str, path)
    # ファイル書き込みする。返り値は成功したか
```

ファイル読み込みは `File.open` を `r` モードで開く。（`openFile` 関数）そして、読み込んで、文字列を返そう。

1.2.1. 易 `openFile` のヒント

- `File#read` を使おう。

1.2.2. 易 `writeFile` のヒント

- `File#read` を使おう。

1.3. 各処理を作っていこう

各コマンドに対応する処理を作っていこう。

```
// 全て start は始まりの行, e は終わりの行です！
printLines(str, start, e)
    // 内容を表示する。
appendLines(str, start, e)
    // 追加する。e は無視しよう。
removeLines(str, start, e)
    // 削除する。
```

1.3.1. 易 `printLines` のヒント

- `char.each` ループと `putc` で1文字ずつ処理していこう。
- `\n` が来たら、次の行に移った合図だ。

1.3.2. 難 `appendLines` のヒント

- `printLines` と同様に行をカウントしていこう。

- 挿入は `String#insert` を使おう.

1.3.3. 易 `removeLines` のヒント

- `printLines` と同様に行をカウントしていこう.

1.4. コマンドを解析しよう

コマンドの範囲選択のところを解析する関数を作ろう.

```
parse(str) // 返り値は始まりと終わり
```

1.4.1. 易 ヒント

- 正規表現を使ってみよう.

1.5. `main` を用意しよう

ここまでできた君ならば行けるはず!