viniciusAbrantes / Sistema-Academia Public

Code Issues Pull requests Actions Projects Security Insights

Commit

⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Extract Method #3

Browse files

MateusAra committed on Jun 19

1 parent 05c0c24 commit c5aedea

Showing **2 changed files** with **56 additions** and **58 deletions**.

Whitespace Ignore whitespace Split Unified

BIN +160 Bytes (100%) Academia Master/build/classes/control/Validador.class

Binary file not shown.

114 Academia Master/src/control/Validador.java

```
363 363                 label.setText(mensagem);
364 364             }
365 365         }
366     -
367     -     private static boolean isValidCpf(String cpf){
368     -
369     -         if (cpf.equals("00000000000") ||
370     -             cpf.equals("11111111111") ||
371     -             cpf.equals("22222222222") ||
372     -             cpf.equals("33333333333") ||
373     -             cpf.equals("44444444444") ||
374     -             cpf.equals("55555555555") ||
375     -             cpf.equals("66666666666") ||
376     -             cpf.equals("77777777777") ||
377     -             cpf.equals("88888888888") ||
378     -             cpf.equals("99999999999") ||
379     -             cpf.length() != 11)
380     -             return(false);
381     -
382     -         char dig10, dig11;
383     -         int sm, i, r, num, peso;
```

```
384 -
385 -         try {
386 -             // Calculo do primeiro Digito Verificador
387 -             sm = 0;
388 -             peso = 10;
389 -             for (i=0; i<9; i++) {
390 -                 num = (int)(cpf.charAt(i) - 48);
391 -                 sm = sm + (num * peso);
392 -                 peso = peso - 1;
393 -             }
394 -             r = 11 - (sm % 11);
395 -             if ((r == 10) || (r == 11))
396 -                 dig10 = '0';
397 -             else
398 -                 dig10 = (char)(r + 48);
399 -
400 -             // Calculo do segundo Digito Verificador
401 -             sm = 0;
402 -             peso = 11;
403 -             for(i=0; i<10; i++) {
404 -                 num = (int)(cpf.charAt(i) - 48);
405 -                 sm = sm + (num * peso);
406 -                 peso = peso - 1;
407 -             }
408 -             r = 11 - (sm % 11);
409 -             if ((r == 10) || (r == 11))
410 -                 dig11 = '0';
411 -             else
412 -                 dig11 = (char)(r + 48);
413 -
414 -             if ((dig10 == cpf.charAt(9)) && (dig11 == cpf.charAt(10)))
415 -                 return(true);
416 -             else return(false);
417 -         } catch(Exception e) {
418 -             return(false);
419 -         }
420 -     }
421 366
422 367       private static boolean isInvalidCnpjDigits(String cnpj) {
423 368           return cnpj.equals("00000000000000") ||
432 377               cnpj.equals("99999999999999");
433 378       }
434 379
435 -     private static char calculateDigit(String cnpj, int position) {
    380 +     private static char calculateDigitCnpj(String cnpj, int position) {
436 381           int sm = 0;
437 382           int peso = 2;
438 383           int num;
467 412           }
468 413
469 414           try {
470 -             char dig13 = calculateDigit(cnpj, 12);
```

```
471     -             char dig14 = calculateDigit(cnpj, 13);
        415 +         char dig13 = calculateDigitCnpj(cnpj, 12);
        416 +         char dig14 = calculateDigitCnpj(cnpj, 13);
472     417
473     418               return (dig13 == cnpj.charAt(12)) && (dig14 == cnpj.charAt(13));
474     419           }
477     422           }
478     423       }
479     424
        425 +     private static char calculateDigitCpf(String cpf, int position) {
        426 +         int sm = 0;
        427 +         int peso = position + 1;
        428 +         int num;
        429 +
        430 +         for (int i = 0; i < position; i++) {
        431 +             num = cpf.charAt(i) - '0';
        432 +             sm += num * peso;
        433 +             peso--;
        434 +         }
        435 +
        436 +         int r = 11 - (sm % 11);
        437 +         char digit;
        438 +         if (r == 10 || r == 11) {
        439 +             digit = '0';
        440 +         } else {
        441 +             digit = (char) (r + '0');
        442 +         }
        443 +
        444 +         return digit;
        445 +     }
        446 +     private static boolean isInvalidCpfDigits(String cpf) {
        447 +         return cpf.equals("00000000000") ||
        448 +             cpf.equals("11111111111") ||
        449 +             cpf.equals("22222222222") ||
        450 +             cpf.equals("33333333333") ||
        451 +             cpf.equals("44444444444") ||
        452 +             cpf.equals("55555555555") ||
        453 +             cpf.equals("66666666666") ||
        454 +             cpf.equals("77777777777") ||
        455 +             cpf.equals("88888888888") ||
        456 +             cpf.equals("99999999999");
        457 +     }
        458 +
        459 +     private static boolean isInvalidCpfFormat(String cpf) {
        460 +         return cpf.length() != 11;
        461 +     }
        462 +
        463 +     private static boolean isValidCpf(String cpf) {
        464 +         if (isInvalidCpfFormat(cpf) || isInvalidCpfDigits(cpf)) {
        465 +             return false;
        466 +         }
        467 +
```

```
468 +         try {
469 +             char dig10 = calculateDigitCpf(cpf, 9);
470 +             char dig11 = calculateDigitCpf(cpf, 10);
471 +
472 +             return (dig10 == cpf.charAt(9)) && (dig11 == cpf.charAt(10));
473 +         }
474 +         catch (Exception e) {
475 +             return false;
476 +         }
477 +     }
480 478   }
```

0 comments on commit `c5aedea`

Please sign in to comment.