



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
Curso de Tecnologia em Sistemas de Computação  
Disciplina: Programação com Interfaces Gráficas  
AD1 1º semestre de 2020.  
Professores: Mario Benevides e Paulo Roma

---

## AD1: DNA Strands

### 1 Informações gerais

Este trabalho é individual. Se precisar de ajuda, procure o seu tutor presencial ou à distância. Ajuda também pode ser obtida na plataforma.

A nota desse trabalho é automática. Se o trabalho não for submetido corretamente, ela será reduzida à metade.

### 2 Objetivos

Este é um exercício para escrever laços e construir strings.

Para esse trabalho, você implementará uma classe chamada DNAStrand, que é um modelo simples de uma fita de DNA. Para nossos propósitos, uma fita de DNA é essencialmente uma sequência de caracteres  $A$ ,  $G$ ,  $C$  ou  $T$ . Dizemos que o caractere  $A$  corresponde a  $T$  e o caractere  $G$  corresponde a  $C$ . Mais tecnicamente,  $A$  forma um par base com  $T$  e  $G$  forma um par base com  $C$ .

Dadas duas cadeias de DNA, elas tendem a se alinhar entre si, para que o número de correspondências seja maximizado. Aqui está um exemplo. Suponha que comecemos com a fita representada pela string

T C A T

e queremos examinar os alinhamentos possíveis com uma outra fita de DNA,

A G A G C A T

Se alinhá-las como

T C A T  
A G A G C A T

então o número de casamentos é dois (T com A, e C com G)

```

| T C | A T
| A G | A G C A T

```

Mas se deslocarmos a segunda fita um espaço para a direita,

```

T C A | T |
A G | A | G C A T

```

então o número de casamentos é apenas um (T com A).

Se deslocarmos dois espaços para a direita,

```

T C A T
A G A G C A T

```

então não há casamentos, e se deslocarmos três para a direita,

```

T C A | T |
| A | G A G C A T

```

há um casamento (T com A).

Poderíamos também deslocar a segunda fita para a esquerda:

```

T C A T
A G A G C A T

```

onde não há casamentos. Se deslocarmos duas casas para a esquerda,

```

| T C | A | T |
A G | A G | C | A | T

```

há três casamentos. Poderíamos também deslocar três para a esquerda,

```

T C A T
A G A G C A T

```

e nesse caso não haveria casamentos, ou por 4,

```

T C | A | T
A G A G C A | T |

```

com um casamento; por 5,

```

| T | C A T
A G A G C | A | T

```

com um casamento; ou por 6,

```

T C A T
A G A G C A T

```

onde não há casamentos.

Assim, o número máximo possível de correspondências, para todos os alinhamentos possíveis das duas fitas, é 3 (que ocorre quando a segunda fita é deslocada para a esquerda por 2).

Também é útil identificar os caracteres nos quais as correspondências ocorrem. Uma notação possível seria representar as letras não correspondentes em minúsculas. Por exemplo, com a sequência TCAT especificada e o alinhamento

```

      | T C | A | T |
A G | A G | C | A | T

```

poderíamos representar os casamentos formados em TCAT usando a string "TCaT".

### 3 A classe DNAStrand

A classe DNAStrand apenas encapsula uma sequência de caracteres e possui várias operações descritas abaixo. Você precisará apenas de uma variável de instância do tipo String. Seu código deve estar no diretório AD1. O construtor e os métodos necessários podem ser encontrados na plataforma online <sup>1</sup>.

#### Notas

1. Lembre-se de que qualquer método do DNAStrand pode acessar as variáveis privadas de instância de outra instância de DNAStrand. Por exemplo, se sua implementação tiver uma variável de instância chamada `data`, um método como `countMatchesWithRightShift(self, other)` pode acessar essa variável de instância como `other.data`.
2. Você pode obter um caractere em uma determinada posição de uma string com `ch = umaStr[index]`. Para converter um caractere `ch` em minúscula e adicioná-lo a uma string de resultados usando o método `lower()`, basta fazer:

```
result += c.lower()
```

#### Sugestão

Elabore alguns exemplos manualmente, usando lápis e papel, antes de começar a codificar e mantenha um controle cuidadoso dos índices. Por exemplo, neste cenário, em que a outra fita é deslocada para a direita por 2:

```

  T C A T
    A G A G C A T

```

Em que índice você pode começar a olhar para o TCAT? Em que índice você precisa parar? Como isso pode ser calculado a partir do comprimento e deslocamentos das strings? E esse aqui, deslocado para a esquerda por 5?

```

      | T | C A T
A G A G C | A | T

```

---

<sup>1</sup><http://orion.lcg.ufrj.br/python/ADs/DNAStrand-skeleton.py>

## 4 Testes

Python possui um mecanismo de automatização de testes muito similar ao JUnit <sup>2 3</sup> do Java. Chama-se unittest <sup>4 5 6</sup>.

Portanto, certifique-se de escrever uma classe DNAStrandTest para testar todos os métodos da sua classe DNAStrand.

```
#!/usr/bin/env python
# coding: UTF-8
#
## @package DNAStrandTest
#
# Class for testing the DNA strand matching.
#
# @author Paulo Roma
# @since 18/12/2019
# @see https://docs.python.org/2/library/unittest.html
#

from DNAStrand import DNAStrand
import sys
import unittest

##
# Class for testing certain aspects of the behavior of
# DNAStrand.
#
class DNAStrandTest(unittest.TestCase):
    ##
    # Two DNAStrand objects to be used in the tests.
    #
    def setUp(self):
        self.d1 = DNAStrand("TCAT")
        self.d2 = DNAStrand("AGAGCAT")

    def test_isValid(self):
        msg = "isValid() should return the given boolean"
        self.assertTrue(self.d1.isValid(), msg)
        self.assertTrue(self.d2.isValid(), msg)

    # .....

if __name__=="__main__":
    unittest.main()
```

---

<sup>2</sup><https://www.devmedia.com.br/junit-tutorial/1432>

<sup>3</sup><https://www.guru99.com/junit-tutorial.html>

<sup>4</sup><https://docs.python.org/3/library/unittest.html>

<sup>5</sup><https://realpython.com/python-testing/>

<sup>6</sup><https://pymotw.com/2/unittest/>

## 5 Documentação e estilo

Já que esse trabalho é simples, a correção será automática e na maioria dos casos o seu código não será olhado. Portanto, não há requisitos específicos de documentação e estilo. No entanto, todos os métodos e seus argumentos devem ser documentados de um modo compatível ao Doxygen<sup>7</sup>.

O Doxygen, além de ser um código aberto, é capaz de produzir documentação para programas escritos em diversas linguagens, como C, C++, C#, Objective-C, Java, Python, Fortran, Perl, PHP, Tcl e VHDL.

Para tanto, baixe o pacote disponível para Linux, MacOS ou Windows, no próprio site do desenvolvedor<sup>8</sup>, e disponibilize no diretório do projeto, um arquivo de configuração chamado Doxyfile (pode ser criado pelo comando **doxygen -g**).

A maioria dos parâmetros contidos no Doxyfile pode permanecer com os valores default, a menos de:

1. PROJECT\_NAME - nome do projeto
2. OUTPUT\_DIRECTORY - diretório no qual será colocada a documentação produzida.
3. INPUT - diretório que será varrido a procura de arquivos a serem documentados.
4. FILE\_PATTERNS - extensões de arquivos a serem documentados, por ex, \*.py.
5. RECURSIVE - se subdiretórios devem ser percorridos recursivamente, em busca de arquivos a documentar.
6. SOURCE\_BROWSER - indica se o fonte dos programas deve ser incluído.

Normalmente, eu uso um diretório chamado doc, dentro do diretório do projeto, para conter a documentação. Após a execução do comando **doxygen**, serão criados três subdiretórios: html, latex e man.

A página html gerada pode ser acessada, apontando o seu browser para a seguinte URL: *file:///proj\_path/doc/html/index.html*, onde /proj\_path é o caminho para o diretório raiz do seu projeto.

A documentação latex pode ser gerada, utilizando o comando **make**, a partir do subdiretório latex. Será criado então um arquivo chamado refman.pdf. É claro que deve-se ter o latex<sup>9</sup> devidamente instalado e configurado para essa opção funcionar.

## 6 Se tiver dúvidas

Para perguntas, consulte a sala de tutoria na plataforma. Se sua pergunta não for respondida, crie uma nova postagem com ela. Tente indicar claramente a pergunta ou o tópico no título da sua postagem e anexe a tag AD1. Mas lembre-se, não publique nenhum código-fonte para

---

<sup>7</sup><http://www.doxygen.nl/manual/>

<sup>8</sup><http://www.doxygen.nl/download.html>

<sup>9</sup><https://www.latex-project.org/get/>

as classes a serem entregues. Não há problema em postar código-fonte para exemplos gerais de Python que não estão sendo entregues.

Se você tiver uma pergunta que absolutamente não possa ser feita sem mostrar parte do seu código-fonte, envie a pergunta de forma privada para que somente os tutores possam vê-la. Certifique-se de fazer perguntas específicas; solicitações vagas tipo "leia todo o meu código e me diga o que há de errado com ele" geralmente serão ignoradas.

Obviamente, os tutores estão sempre disponíveis para ajudá-lo. Consulte o horário de tutoria para encontrar um horário conveniente para você. Faremos o possível para responder todas as perguntas com cuidado, a menos de realmente escrever o código por você, pois seria injusto para a equipe revisar sua tarefa em detalhes antes dela ser entregue.

Quaisquer postagens dos instrutores na plataforma intituladas "Esclarecimento oficial" são consideradas partes da especificação e você pode perder pontos se ignorá-las. Essas postagens sempre serão colocadas na plataforma, além da sala de tutoria. (Prometemos que nenhum esclarecimento oficial será publicado antes de 24 horas da data de entrega.)

## 7 O que entregar

Por favor, envie, na plataforma, o arquivo zip criado por você. O arquivo será chamado NOME\_SOBRENOME\_AD1\_PIG.zip e ele estará localizado no diretório que você selecionar ao executar o programa Zip. Ele deve conter um diretório, AD1, que por sua vez contém os arquivos chamados:

**DNAStrand.py, DNAStrandTest.py e refman.pdf.**

O arquivo comprimido deve ser um arquivo zip; verifique se você está usando o utilitário zip do Linux, Windows ou Mac, e não uma implementação de terceiros do WinRAR, 7-zip ou Winzip.

## 8 Considerações finais

Documentação e testes são talvez as partes mais importantes e demoradas de um projeto. Um projeto não documentado é inútil. Em pouco tempo perde-se todo o histórico do seu desenvolvimento.

Para que o produto possa ser utilizado globalmente, é fundamental que nomes de classes, métodos, parâmetros e variáveis, assim como todos os comentários, estejam em inglês. A não ser que o conhecimento da língua inglesa seja muito limitado, essa é a decisão acertada.

A documentação deve ser escrita durante o desenvolvimento e não apenas ao final do projeto. O ato de documentar nos força a repensar o porque de determinadas decisões e garante que referências importantes não sejam esquecidas.