UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

1ª Avaliação

O Sistema ABO é um sistema que classifica o sangue em quatro diferentes tipos, os quais se diferenciam pelas suas aglutininas e aglutinogênios. Este sistema classifica grupos sanguíneos em tipos A, B, AB e O.

Os grupos sanguíneos ABO são determinados por três alelos diferentes de um único gene: I^A, I^B e i. Esses três alelos são os responsáveis por garantir na espécie humana a presença de quatro fenótipos: sangue A, sangue B, sangue AB e sangue O.

O sistema ABO é um exemplo clássico de alelos múltiplos e de codominância. É um caso de alelos múltiplos, pois apresenta três alelos diferentes de um único gene (I^A, I^B e i). A codominância também ocorre, pois entre os alelos I^Ae I^B não existe relação de dominância. Com isso temos que os genótipos dos tipos sanguíneo são:

Tipo Sanguíneo	Genótipo
A	I^AI^A ou I^Ai
B	I^BI^B ou I^Bi
AB	I^AI^B
O	ii

O quadro de Punnett é uma tabela em que é possível separar os possíveis gametas e descobrir os genótipos dos descendentes. Suponha que o tipo sanguíneo de cada pessoa de um casal seja O (ii) e AB (I^AI^B), portanto os possíveis tipos sanguíneos de seus descendentes seriam A ou B, conforme ilustrado no quadro abaixo:

	i	**i**
I^A	I^Ai	I^Ai
I^B	I^Bi	I^Bi

Considerando-se apenas o sistema ABO, pode-se fazer algumas afirmações importantes em relação à transfusão de sangue:

Tipo	Doam Para	Recebem De
A	A, AB	A, O
B	B, AB	B, O
AB	AB	A, B, AB, O
O	A, B, AB, O	O

Veja mais sobre "Sistema ABO" em: https://brasilescola.uol.com.br/biologia/sistema-abo.htm

Com base nesses conceitos elementares sobre o Sistema ABO, desenvolva uma aplicação em Python 3 que implemente os seguintes requisitos e funcionalidades:

REQUISITOS E FUNCIONALIDADES:

1. Implemente a classe "*Individual*" que representa um indivíduo com determinado genótipo de tipo sanguíneo.
    - *Um indivíduo é representado pelo genótipo de seu tipo sanguíneo (obrigatório) e seu nome (opcional).*
    - *O genótipo deverá ser especificado por uma das strings: "AA", "Ai", "BB", "Bi", "AB" ou "ii", onde os alelos "A" e "B" dominam o alelo "i".*
    - *A classe deverá levantar uma exceção no caso de genótipo inválido.*
    - *Todos os atributos da classe deverão ser privados.*
2. Caso o construtor da classe "*Individual*" receba apenas o tipo sanguíneo, o nome do indivíduo deverá ser gerado automaticamente no formato "IndivN", onde "N" é um número inteiro sequencial que inicial por 1.
    - *Não podem existir indivíduos com o mesmo nome gerado automaticamente.*
    - *Não é permitido o uso de variáveis globais para controle do sequenciamento dos nomes automáticos.*
3. Implemente suporte para a conversão para string de uma instância de "*Individual*".
4. Implemente a propriedade *read-only* "*name*" que retorna o nome do indivíduo.
5. Implemente a propriedade *read-only* "*genotype*" que retorna o genótipo do indivíduo.
6. Implemente a propriedade *read-only* "*blood_type*" que retorna o tipo sanguíneo do indivíduo.
7. Implemente a propriedade *read-only* "*agglutinogens*" que retorna os tipos de aglutinogêneos presentes no sangue do indivíduo.
    - *Os aglutinógenos devem ser identificados por "A" e "B"*
8. Implemente a propriedade *read-only* "*agglutinins*" que retorna os tipos de aglutininas presentes no sangue de um indivíduo.
    - *As aglutininas "anti-A" e "anti-B" devem ser identificadas por "A" e "B", respectivamente*
9. Implemente o método "*offsprings_genotypes*" que retorna uma sequência dos possíveis genótipos resultantes do cruzamento entre o indivíduo que chama o método e outro indivíduo passado como parâmetro.
    - *Não deve haver repetição de genótipos na sequência resultante do cruzamento.*
10. Implemente o método "*offsprings_blood_types*" que retorna uma sequência dos possíveis tipos sanguíneos resultantes do cruzamento entre o indivíduo que chama o método e outro indivíduo passado como parâmetro.
    - *Não deve haver repetição de tipos sanguíneos na sequência resultante do cruzamento.*
11. Implemente o método "*can_donate*" que verifica se o indivíduo que chama o método pode doar sangue para outro indivíduo passado como parâmetro.
12. Implemente o método "*can_receive*" que verifica se o indivíduo que chama o método pode receber sangue de outro indivíduo passado como parâmetro.
13. Seu código será testado com a função "*main()*" existente no arquivo "main.py", que poderá ser baixado do repositório GitHub disponibilizado na seção "Recursos do Projeto" apresentada adiante.
    - *O código do arquivo "main.py" não deverá ser alterado. Seu código deve se adaptar às instruções lá existentes.*
14. A codificação do projeto deverá observar os princípios de estilo do PEP8 (https://peps.python.org/pep-0008).
15. O projeto deverá ser disponibilizado em um repositório com acesso público do GitHub, com "main" como nome do seu *branch* principal (*default* do GitHub).
    - *Qualquer outro nome de branch não será considerado.*

RECURSOS DO PROJETO:

1. Todos os arquivos necessários para o projeto estão disponíveis no repositório https://github.com/uespi-phb/python-blood.git
2. Os arquivos disponibilizados não deverão ser alterados.

ARQUIVO	DESCRIÇÃO
main.py	Arquivo principal para teste do código submetido.
output.txt	Saída esperada ao executar o programa a partir do arquivo "main.py"

COMPOSIÇÃO DA EQUIPE:

1. A implementação poderá ser realizada em equipes de ATÉ 2 (dois) membros.
2. Os autores de cada implementação poderão ser questionados sobre o código implementado, com o objetivo de comprovar a participação de cada membro na execução do projeto.

INTRUÇÕES PARA REMESSA DO PROJETO:

1. Ao finalizar o projeto, remeter o *link* do repositório GitHub para o e-mail: eyder@phb.uespi.br com o seguinte assunto: "PROG1 – AVAL1", juntamente com os nomes dos membros da equipe no corpo do e-mail;
2. Prazo de entrega: 16/12/2022;
3. O projeto deverá ser compatível com a versão 3 da linguagem Python.

CRITÉRIOS DE AVALIAÇÃO:

- Cada requisito/funcionalidade será avaliado individualmente e receberá uma das seguintes notas:

Nota	Grau de atendimento ao requisito/funcionalidade
0	Não atende
1 a 4	Atende parcialmente
5	Atende completamente

- A nota final será determinada pela média ponderada das notas de cada requisito/funcionalidade.