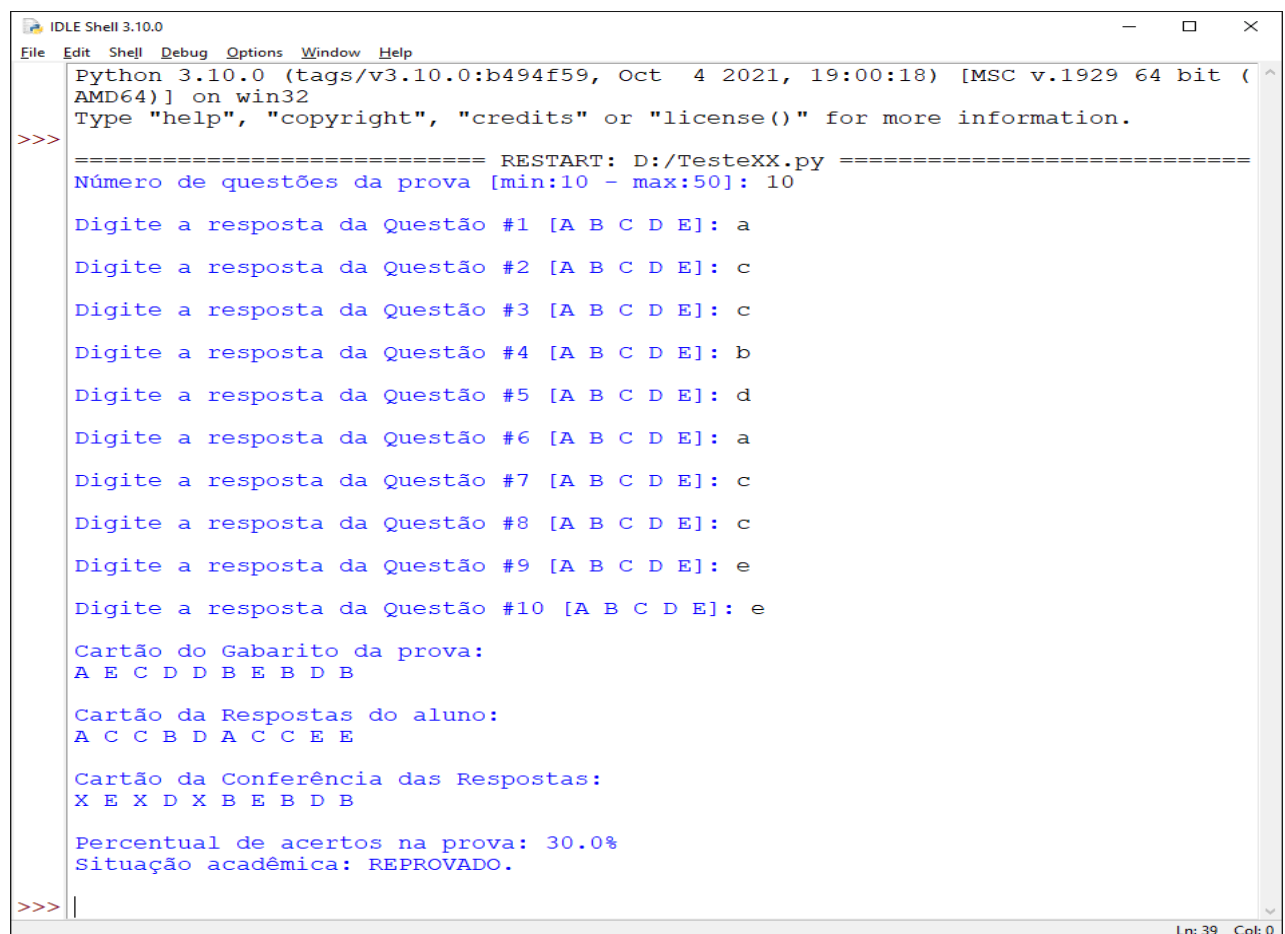


Conferindo Prova

Mário Leite

...

Há muito tempo atrás as provas eram, todas, subjetivas: os alunos tinham que demonstrar todas as respostas de todas as questões. Assim, ficava comprovado que aquela resposta dada àquela questão estava embasada nos conhecimentos adquiridos nos seus estudos e media, efetivamente o que ele havia aprendido nas aulas expostas pelos seus professores. Com o advento da massificação do ensino e com a proliferação desenfreada de cursos “superiores” (muitos apenas por conveniência financeira) as provas e testes escolares passaram a ser ditos “objetivos” (aspas intencionais), mas nem sempre favorável à absorção de conhecimento. Assim, surgiram as chamadas provas com questões de múltiplas escolhas, fazendo com que o aluno não precisasse demonstrar como chegou àquela resposta daquela questão, pois a conferência nem precisava mais de professores da disciplina; a conferência das questões passou a ser através de um GABARITO com as repostas corretas, confrontando-as com as sequências de repostas marcadas pelo aluno. Deste modo qualquer um agora pode conferir as provas de centenas de alunos de uma maneira rápida e sem discussões (não há mais necessidade de atender reclamações de aluno para revisar sua prova). Este processo, embora muito contestado por professores (eu me incluo), é uma realidade na prática. Assim, resolvi criar um programa que confere as respostas de uma prova de múltipla escolha, usando um gabarito gerado aleatoriamente para confrontar com as respostas do aluno. A **figura 1** mostra o gabarito de uma prova, juntamente com as respostas de um o aluno. Neste caso, vê-se claramente que esse aluno “chutou” as respostas, pois acertou apenas três questões em dez propostas. O programa “**ConfereProva**” é uma solução simples, baseada na orientação a objetos, criando uma classe que encapsula *atributos e métodos* do elemento “Aluno”.



```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/TesteXX.py =====
Número de questões da prova [min:10 - max:50]: 10

Digite a resposta da Questão #1 [A B C D E]: a
Digite a resposta da Questão #2 [A B C D E]: c
Digite a resposta da Questão #3 [A B C D E]: c
Digite a resposta da Questão #4 [A B C D E]: b
Digite a resposta da Questão #5 [A B C D E]: d
Digite a resposta da Questão #6 [A B C D E]: a
Digite a resposta da Questão #7 [A B C D E]: c
Digite a resposta da Questão #8 [A B C D E]: c
Digite a resposta da Questão #9 [A B C D E]: e
Digite a resposta da Questão #10 [A B C D E]: e

Cartão do Gabarito da prova:
A E C D D B E B D B

Cartão da Respostas do aluno:
A C C B D A C C E E

Cartão da Conferência das Respostas:
X E X D X B E B D B

Percentual de acertos na prova: 30.0%
Situação acadêmica: REPROVADO.
>>>
```

Figura 1 - Conferindo as respostas de um aluno com um Gabarito

```
'''
ConfereProva.py
-----
Confere as respostas de um aluno numa prova de múltipla escolha, usando
um Gabarito gerado aleatoriamente, para conferir as respostas do aluno.
-----
'''

import random

class ClsAluno:
    def __init__(self):
        self.LstMatResp = []
        self.LstMatConf = []
        self.acertos = 0
        self.percAcerto = 0.0

    #-----
    def LerRespostas(self, n):
        print("")
        for i in range(1, n+1):
            while True:
                resp = input(f"Digite a resposta da Questão #{i} [A B
                C D E]: ").upper()
                if (resp in ['A', 'B', 'C', 'D', 'E']):
                    break
            self.LstMatResp.append([resp])
            print("")

        #Cria uma matriz de conferência
        for i in range(1, n+1):
            self.LstMatConf.append([LstMatGaba[i - 1][0]])

    #-----
    def ConferirRespostas(self, n, LstMatGaba):
        self.acertos = 0
        for i in range(1, n+1):
            if (self.LstMatResp[i-1][0]==LstMatGaba[i-1][0]): #acertou
                self.LstMatConf[i-1][0] = "X"
                self.acertos += 1
            else: #errou a questão
                self.LstMatConf[i-1][0] = LstMatGaba[i-1][0]

        #Exibe os resultados
        print("Cartão do Gabarito da prova:")
        print(" ".join([LstMatGaba[i-1][0] for i in range(1, n+1)]))
        print("\nCartão da Respostas do aluno:")
        print(" ".join([self.LstMatResp[i-1][0] for i in range(1,n+ 1)]))
        print("\nCartão da Conferência das Respostas:")
        print(" ".join([self.LstMatConf[i-1][0] for i in range(1,n+ 1)]))
        print("")

    #-----
    def MostraResultados(self, n, PERC_MIN):
        self.percAcerto = (self.acertos / n)*100 #percentual de acertos
        self.percAcerto = round(self.percAcerto, 1)
        print(f"Percentual de acertos na prova: {self.percAcerto}%")

```

```

        if self.percAcerto >= PERC_MIN:
            print("Situação acadêmica: APROVADO.")
        else:
            print("Situação acadêmica: REPROVADO.")

        print("")

=====
#Programa principal
MAX_QUEST = 50    #define o máximo de questões da prova
PERC_MIN = 60     #percentual mínimo de acertos para aprovação
n = 0
while(True):
    try:
        n = abs(int(input(f"Número de questões da prova [min:10
        - max:{MAX_QUEST}]: ")))
    except ValueError:
        pass

    if(10 <= n <= MAX_QUEST):
        break

LstMatGaba = [[random.choice('ABCDE')] for _ in range(1, n+1)]
Aluno = ClsAluno()    #cria instância da classe "ClsAluno"
Aluno.LerRespostas(n)
Aluno.ConferirRespostas(n, LstMatGaba)
Aluno.MostraResultados(n, PERC_MIN)
#Fim do programa "ConfereProva" -----

```