Tecrologia En sistemas de Comp Clamber de South FAR. A 17213050160 Angra dos Rais AP2X-PROG. COM-INERFACES

GRAGECAS

```
DEF F(A,B):

IF A == [] : RETURN []

IF A [0] IN B: RETURN F [A[1:],B)

RETURN F [A[1:],B) + A [:1]

PRINT (F(LIST (RANGE (10)), [1,9]))
```

```
FROM OPERATOR IMPORT ITEM GETTER
Class PROVA (ObjeKT):
   DEF _ - INIT - - (SElt, NOME) =
       Self. Nome = Nome
       SOLF. PROVA = []
   DEF RESP (SELF, Q, 1):
       IF (SEIF- VERIFICA Q(Q)):
           IF (SELF. VERIFICAL (1)):
              IF (SUF. VERIFICA EVITE (D(Q)):
                 Selfa Substitui (Q, L)
              Else:
                RESPOSTA = ["Q" : Q, "/":/]
                Self. prove - Append (RESPOSTA)
         Elsei:
            PRINT ("1"+ L+" NÃO E UN VALOR VATISTO PARA 1")
      ESE:
         PRINT ("Value ERROR, QNÃO ENSTURAL de 1 + 25")
DEF Substitui (self, Q, L):
    FOR I IN RANGE (O, LEN(SelF. prova)):
        IF Q == SELF. PROVACIJE"Q"]:
            SelF. PROVA [I] [ Q"] = Q
            Self. PROVA [ ] [ " L"] = L
            BREAK
```

```
DEF VERIFICAEXISTE Q (SEIF, Q):
     Isvalid = FalsE
      FOR I IN RANGE (O, LEN (SELF. PRO VA)):
         IF Q = = Self - PROVA [ : ] ["Q"] .
             Is Valid = TRUE
         IF IsValid:
            RETURN TRUE
         Else:
             RETURN FAISE
 DEF VERIFICAQ (SELF, Q):
   Is Valid = Felse
   For I IN Range (1, 26):
       TF Q = II:
            Is valid = TRUE
   IF ISVALID:
      RETURN TRUE
  Else:
       RETURN FAISE
PET VERIFICAL (SEIF, L):
  Alternativa = "ABCD"
  IsValid = FAISE
  FOR I IN AlTZENATIVAS:
       IF L == I:
            IsValid = TRUE
  IF INALID: BREAK
  Else RETURN FAISE
```

LEF -- REPR -- (SEIF): RETURN STR (SELF. NOME) + " = "+STR (Self. MAPUNE FOR (SIL) 4 SEIF OPCHENIA PROVA()) DEF DADENA PROVI (SEIF): RETURN SORTEd (SOLF, PROVA, KEY = ITEMBETTER ('Q')) DEF ARRUMA PROVA (SELF, PROVA ORDENADA) = QUESTOES = 1111 FOR I IN RANGE (O, LEN (PROVA ORDENADA)) : IF (I! = LEN(PROVA Dedervada) -1): QUESTOES += STR (PROVA ORDENADA [:]["Q"]+'-'+ & STR (PROVA ORDENADO EIJEMLIJ)+1 A Else: QUESTOEST = STR (PROVA Dedenoda [i] ["Q"]+'-1'+ STR (PROVA ORDENA de [3[1])+11 RETURN QUESTOES

```
CLASS GABARITO (PROVA)=
   DEF_- INIT -- (SEIF):
      SELF. PROVA = []
  DEF Resp (self, Q, L):
      IF (Self. VERIFICA (RQ)):
         IF (SEIF-VERIFICAL (L)) =
            IF (SELF. VERIFICA ExiSTE Q(Q)):
               SELFO VARIAS RESPOSTAS (Q, 1)
            Else:
               REPOSTA = {"Q": Q, "L": L}
               SELF. PROVA. Append (RESPOSTA)
        Else:
         PRINT (10, 11+2+11, NÃO E UM VATOR VATIRO PARA L'11)
    Else:
      PRINT ("VA lut ERROR, Q. NãO + MATURAL de 1 4 28")
DEF VARIAS RESPOSTAS (SEIF, Q, L):
    FOR I IN RANGE (O, LEN (SELF. PROVA)):
     If Q == SEIF. PROVAEIJ ['Q']
         SELF- PROVA EIJE"1"] = [Self. PROVA []["L"], 2]
        BREAK
 DEF VERIFICA Q (SEIF, Q):
    Isvalid = FAISE
    FORIIN RANGE (1, 26):
       IF O == I:
         Is Valid = TRUE
   IF ISVALID!
       RETURN TRUE
   E SE
      RETURN FAISE
```

DEF VERIFICAL (SELF, L) = ALTERNATIVAS = 'ABCDX IsVAlid= FAlsE FOR I IN AlteRNATIVAS: IF L == I; ISVA/: D=TRUE BREAK IF Is VAlid: RETURN TRUE Else: RETURN FALSO DEF VERIFICA EXITE Q (SELF, Q) : Is Valid = FA/St FOR I IN RANGE (0, LEN (SOLF. PROVE)): If Q == Self. PROVA [i] ["O"]: Is Valid = TRUE BREAK IF Is Valid= RETURN TRUE Else: RETURN FAISE Def ORDENA CABORITO (SEIF) RETURN SORTEd (Self-PROVA, Key: I TEMPETER ('Q'))

DEF APRUMA GABARITO (SEIF, GABARITO ORCHANACIO): QUESTOES = 11 11 FOR I IN RANGE (O, LAN (gabaRito ORDENAdo): IF (Lew (gabarito Ordonado Ei] ["L"]) >1):

Questoes += 512 (gabarito Ordenach ti] ['Q']) + 1-1

FOR J IN Range (Lew (gabarito Ordenado [i] ['L'])): IF(j!= Len (gabae: TO Ordensdo [i]['2'])+1): QUESTOES + = STAL gabarTo Oackrado [i]['2][j])+" Questoes += sta (gaber: Po Oademoto [: 2["2][]]) QUESTOES. += 1 IF(I!= Len (gabacico Ordenado)-1): Questoes += STR(gabacico Ordenado [:]['a']+!-!+ STA (gabarito Ordensto 5: 2523)+1 Else: QUEITOCS += STR (gabarito Opdersodo [i]['Q']+'-'+ STA (gabreiro Dedonado Ei]['L'])] RETURN QUESTOES

```
DEF NOFA (SEIF, P):
    VALOR QUESTAD = SE) F. VALOR QUESTÃO ()
     ERROS, ACERTOS = SEIF . VERIFICA ACERTOS (P)
     NOTA = (ALERTOS # VALOR QUESTAD) - (ERROS * (VALOR QUESTÃO/2))
     RETURN NOTA
Par VERIFICA DEERTOS (SOIF, P) =
    PUPPOS = O
    ERROS = 0
     Armstadas = D
    FOR I IN RANGE (CO- (SOIF. Paora)) :
       FOR JIN Range (Lew (P. Pasva)):
         IF Solf. Prova [: ] ['Q'] == P. Provo[J]['Q']:
            FOR K IN RANGE (LEN(SOF, PROVAE: 3E'L']):
               IF (SHF- PROVA ET & ['L'] [K] == P. PROVA [J]['L']):
                 ACERTOS += 1
       EliF SelF. PROVA [:]['1'] == P. PROGE [J]['1']:
          ACERTOS += 1
         IF SELF. PROVE [ ] [ 'L'] ! = 1x1:
           ERROS += 1
        Elst:
         Amladas += 1
 RETURN (ERROS, ALERTOS)
```

```
DEF VALOR QUESTAD (SEIF):
       Harladas = 0
       FOR I IN RANGE (Led Stilf. PROVA)):
           IF Self. PROVA [:]['L'] == 'x1;
                Aruladas += 1
       RETURN 10/ (LEN (SEIF. PROVA) - Anoladas)
   DEF __ Repr_ (SEIF):
       RETURN 1 GABARITO 1 + STRIGET PRE uma GABREITO ( A
                                         Self. Orders Embarico())
A = PROVA ('JOHO')
A. RESP (3, 'A'); A. RESP(2, 'B'); A. Resp(4, 'C'); A. Resp(5, 'D')
B= PROVA ( MARTA!)
B. RESP (3, 181); B. RESP (2, 'B'); B. Resp (3, '0'), B. Resp (5, '0')
g: 6abar: (01)
 g. ResP(s,'B'); g. ResP(2,'B'); g. Resp(3,'C'); g. Resp(3,'D')
g. ResP(4, 'A'); g. Resp(s,'X')
TRY 3. RESP(2, 'E')
EXCEPT: VALUE ERROR AS 6:
   PRIN(E)
```

PRINT (g, 4-->", g-NOTA (g)) PRINT (A, "-->", g. NOTO(A)) PRINT (B,"->", g-NOTA(B))