

NOM Prénom

Groupe

SIM-SysIn : Test Final, partie écrite

(Durée 20 minutes, sans document)

1 Extraits du fichier « DiscreteTime.py »

```
1 class DiscreteTime:
2     "models the concept of 'discrete time', useful for digital signal design"
3     def __init__(self, n, step=1.0, start=0.0):
4         self.__to = start          # time origin
5         self.__dt = step           # time step
6         self.__nb = n              # number of discrete values
7         # access methods to read attributes (getters)
8         def start(self): return self.__to
9         def step(self): return self.__dt
10        def nbVal(self): return self.__nb
11        # access methods to write attributes (setters)
12        def setPeriod(self, newStep): self.__dt = newStep
13        # useful methods
14        def delay(self, delayVal): self.__to = self.__to + delayVal
15        def timeValues(self):
16            liste = []
17            for i in range(self.__nb): liste.append(self.__to + i*self.__dt)
18            return liste
19        def t(self, i): return self.__to + (i%self.__nb)*self.__dt
20        def last(self): return self.t(-1)
21        ...
22 if __name__=="__main__":
23     t1 = DiscreteTime(10, 0.1, 1)
24     print "start:", t1.start(), ", step:", t1.step()
25     print "values:", t1.timeValues()
26     print "t(9):", t1.t(9), "last:", t1.last()
```

Question 1 – À quoi sert la ligne 3 ?

Question 2 – Que fait la méthode delay ?

Question 3 – Détailler ce que fait la méthode timeValues.

Question 4 – Que fait la ligne 23 ?

Question 5 – On fait exécuter le fichier « DiscreteTime.py » dans un shell *Python*, que voit-on s'afficher ?

2 Extraits du fichier « DigitalSignal.py »

```

1  from copy import copy
2  from DiscreteTime import DiscreteTime
3  class DigitalSignal(DiscreteTime):
4      "To implement the concept of 'digital signal'"
5      def __init__(self, deltaT=1.0, startT=0.0, values = [0.0]):
6          DiscreteTime.__init__(self, len(values), deltaT, startT)
7          self.__vX = values          # digital signal values (time domain)
8          self.__vZ = None           # Fourier Transform values
9          self.__updateFT()          # computes the Fourier Transform (FT)
10     # access methods to read attributes (getters):
11     def sigValues(self): return copy(self.__vX)
12     def FTvalues(self): return copy(self.__vZ)
13     def parameters(self): return self.step(), self.start()
14     # computed values:
15     def samplingFreq(self): return 1./self.step()
16     def deltaF(self): return 1./(self.step()*self.nbVal())
17     # access methods to write attributes (setters):
18     def setPeriod(self, newP):
19         "To set a new time step value"
20         DiscreteTime.setPeriod(self, newP)
21         self.__updateFT()
22     ...
23 if __name__=="__main__":
24     pas, to, v = 0.1, 0., [0,0,1,0,0,0,0,0,0]
25     ds1=DigitalSignal(pas, to, v)
26     print "parameters:", ds1.parameters()
27     ds1.setPeriod(0.15)
28     print "parameters:", ds1.parameters()

```

Question 6 – À quoi sert la ligne 2 ?

Question 7 – À quoi sert la ligne 3 ?

Question 8 – À quoi sert la ligne 6 ?

Question 9 – En vous aidant de l'extrait de « DiscreteTime.py » au verso, expliquer le calcul de la ligne 16.

Question 10 – Quel est le rôle de la ligne 20 dans la méthode setPeriod.

Question 11 – On fait exécuter le fichier « DiscreteTime.py » dans un shell *Python*, que voit-on s'afficher ?