

Projeto interdisciplinar

Grupo

Guilherme Carini (Ciência da computação) RGM: 20224435

Gabriel pena (Ciência da computação) RGM: 18777651

Ayrton Feliciano (Gestão de TI) RGM: 29641977

Nathan de Oliveira (Gestão de TI) RGM: 3027175

Nikolas Yan da Silva (Gestão de TI) RGM: 29908027

Sobre

É um software de conversão de bases decimais para as respectivas bases binário, octal, hexadecimal, a linguagem de programação utilizada para a construção do mesmo foi, Python ensinada na aula de Técnicas de Desenvolvimento de Algoritmos pelo Professor Genivaldo Silva, e foram aplicados fundamentos ensinando em Organização e Arquitetura de Computadores pelo Professor Álvaro Prado

Descrição do software

O programa se inicia em uma estrutura de repetição (*While*) Onde vai perguntar para o usuário qual das opções quer que o programa execute, “ de decimal para binário, octal e hexadecimal” ou vice-versa, caso a opção for selecionada irá cair em uma estrutura de (*If else*) que irá direcionar para a devida função a ser executada.

```
while True:
    #Menu option list
    print("[1] Converter número decimal para base binária")
    print("[2] Converter número decimal para base octal")
    print("[3] Converter número decimal para base hexadecimal")
    print("[4] Converter base binária para decimal")
    print("[5] Converter base octal para decimal")
    print("[6] Converter base hexadecimal para decimal")
    print("[99] Exibir créditos")
    print("[0] Sair")

    #Menu selected option by user
    menuOption = input("Informe a opção desejada: ")

    #Menu selected option validation for not breaking the software
    if isinstance(menuOption, int) == True:
        menuOption = int(menuOption)

        #Menu workflow according the selected option
        if menuOption == 0:
            exit(0)
        elif menuOption == 1:
            numberValue = input("Informe número a ser convertido: ")
            callingCalculator = Calculator(numberValue, 10, 2)
            callingCalculator.selectWorkflow()
        elif menuOption == 2:
            numberValue = input("Informe número a ser convertido: ")
            callingCalculator = Calculator(numberValue, 10, 8)
            callingCalculator.selectWorkflow()
        elif menuOption == 3:
            numberValue = input("Informe número a ser convertido: ")
            callingCalculator = Calculator(numberValue, 10, 16)
            callingCalculator.selectWorkflow()
        elif menuOption == 4:
            numberValue = input("Informe número a ser convertido: ")
            callingCalculator = Calculator(numberValue, 2, 10)
            callingCalculator.selectWorkflow()
        elif menuOption == 5:
            numberValue = input("Informe número a ser convertido: ")
            callingCalculator = Calculator(numberValue, 8, 10)
            callingCalculator.selectWorkflow()
        elif menuOption == 6:
            numberValue = input("Informe número a ser convertido: ")
            callingCalculator = Calculator(numberValue, 16, 10)
            callingCalculator.selectWorkflow()
        elif menuOption == 99:
```

Após ter escolhido qual das funções executar irá ser redirecionado de main.py para calculator.py onde irá preparar o input do usuário para ser convertido nas devidas bases, primeiro o valor do usuário será convertido de *int* para *string* onde será introduzido a um *array* onde cada caractere será separado dentro desse *array* e cada um será convertido para inteiro novamente, logo após isso sua base será verificada para determinar qual calculo deverá ser feito.

Se a base atual for menor do que dez será aplicado o método de multiplicação caso a base atual for maior do que dez será aplicado método de divisão, chamando o método criado em outra classe do Python que nomeei de utils

```
from Utils import Utils

class Calculator:
    def __init__(self, numberValue, numberBaseCurrent, numberBaseTarget):
        self.numberValue = str(numberValue)
        self.numberBaseCurrent = numberBaseCurrent
        self.numberBaseTarget = numberBaseTarget

        #Split each digit into a different index inside array
        self.arrayStringNormal = list(self.numberValue)
        self.arrayStringNormal = Utils.convertArrayWithLettersToNumbers(self.arrayStringNormal)
        self.arrayNumberNormal = [ int(x) for x in self.arrayStringNormal ]
        self.arrayStringInverted = list(reversed(self.arrayStringNormal))
        self.arrayNumberInverted = [ int(x) for x in self.arrayStringInverted ]

        #self.numberValue = int(self.numberValue)
        self.numberValue = self.numberValue

    def selectWorkflow(self):
        callingUtils = Utils(self.arrayNumberInverted, self.numberValue, self.numberBaseCurrent, self.numberBaseTarget)

        callingUtils.displayAllInfo()

        #Decision structure calculation workflow
        if self.numberBaseCurrent == self.numberBaseTarget:
            result = self.numberValue
        elif self.numberBaseCurrent > 10:
            result = callingUtils.convertBaseMethodMultiplier(self.arrayNumberNormal)
        elif self.numberBaseTarget > 10:
            result = callingUtils.convertBaseMethodDivider(self.numberValue)
        else:
            if self.numberBaseCurrent <= self.numberBaseTarget:
                result = callingUtils.convertBaseMethodMultiplier(self.numberValue)
            elif self.numberBaseCurrent >= self.numberBaseTarget:
                result = callingUtils.convertBaseMethodDivider(self.numberValue)

        #Display results
        callingUtils.displayResult(result)
        return result
```

Dentro da classe Utils foi criado os cálculos de conversão como o de multiplicação que seria:

Algarismo vezes a base numérica desejada pelo usuário sobre sua posição correspondente e esse cálculo será aplicado para cada algarismo, após ter o resultado obtido destas multiplicações, esses resultados serão somados para obter o valor desejado em código ficou:

```
def convertBaseMethodMultiplier(self, numberAux):
    print(numberAux)
    aux = 0
    amount = 0

    for i in range(len(self.array)):
        aux = self.array[i] * (self.numberBaseCurrent ** i)
        amount += aux
        print(f"{self.array[i]} * {self.numberBaseCurrent} ^ {i} = {aux}")

    return amount
```

Agora para o método de divisão seria o valor numérico dividido pela base desejada , o produto de sua divisão irá ser dividido pela base desejada novamente até que seja indivisível e o resto da divisão desses valores numéricos serão multiplicados pela base desejada caso o valor do produto seja indivisível o for irá parar e todos os números que sejam maiores que 9 serão convertidos para o seu respectivo valor em sua determinada base utilizando da função “convertArrayWithNumberstoLetters” assim mostrando o valor desejado

```
def convertBaseMethodDivider(self, numberAux):
    arrayNormal = []
    aux = int(numberAux)
    i = 0
    rest = 0
    divisionInteger = 0

    while aux > 0:
        rest = aux % self.numberBaseTarget
        amount = rest
        arrayNormal.append(amount)
        divisionInteger = aux // self.numberBaseTarget
        aux = divisionInteger
        i += 1

        print(f"{aux} / {self.numberBaseTarget} = {amount}\tRest:\t{rest}")

    arrayInverted = list(reversed(arrayNormal))
    self.convertArrayWithNumbersToLetters(arrayInverted)

    return ''.join(map(str, arrayInverted))
```

```
def convertArrayWithLettersToNumbers(arrayAux):
    i = 0

    #Search each element in list to replace the letter to number value
    while i < len(arrayAux):
        if arrayAux[i] == "A": arrayAux[i] = 10
        elif arrayAux[i] == "B": arrayAux[i] = 11
        elif arrayAux[i] == "C": arrayAux[i] = 12
        elif arrayAux[i] == "D": arrayAux[i] = 13
        elif arrayAux[i] == "E": arrayAux[i] = 14
        elif arrayAux[i] == "F": arrayAux[i] = 15
        elif arrayAux[i] == "G": arrayAux[i] = 16
        elif arrayAux[i] == "H": arrayAux[i] = 17
        elif arrayAux[i] == "I": arrayAux[i] = 18
        elif arrayAux[i] == "J": arrayAux[i] = 19
        elif arrayAux[i] == "K": arrayAux[i] = 20
        elif arrayAux[i] == "L": arrayAux[i] = 21
        elif arrayAux[i] == "M": arrayAux[i] = 22
        elif arrayAux[i] == "N": arrayAux[i] = 23
        elif arrayAux[i] == "O": arrayAux[i] = 24
        elif arrayAux[i] == "P": arrayAux[i] = 25
        elif arrayAux[i] == "Q": arrayAux[i] = 26
        elif arrayAux[i] == "R": arrayAux[i] = 27
        elif arrayAux[i] == "S": arrayAux[i] = 28
        elif arrayAux[i] == "T": arrayAux[i] = 29
        elif arrayAux[i] == "U": arrayAux[i] = 30
        elif arrayAux[i] == "V": arrayAux[i] = 31
        elif arrayAux[i] == "W": arrayAux[i] = 32
        i += 1

    return arrayAux
```

Após a execução do programa utilizando do comando via terminal (Python main.py) selecionando e inserindo os dados desejados será retornado para o usuário todo o processo e execução feita passo a passo da conversão dos números. O usuário poderá fazer novas execuções enquanto o programa estiver dentro da estrutura de repetição (*While*) selecionando a opção desejada poderá finalizar a execução do programa

```
Windows PowerShell
[1] Converter número decimal para base binária
[2] Converter número decimal para base octal
[3] Converter número decimal para base hexadecimal
[4] Converter base binária para decimal
[5] Converter base octal para decimal
[6] Converter base hexadecimal para decimal
[99] Exibir créditos
[0] Sair
Informe a opção desejada: 1
Informe número a ser convertido: 123
Array:          [3, 2, 1]
Número:         123
Base current:   10
Base target:    2
-----
61 / 2 = 1      Rest: 1
30 / 2 = 1      Rest: 1
15 / 2 = 0      Rest: 0
7 / 2 = 1       Rest: 1
3 / 2 = 1       Rest: 1
1 / 2 = 1       Rest: 1
0 / 2 = 1       Rest: 1

Before: [1, 1, 1, 1, 0, 1, 1]

0 = 1
1 = 1
2 = 1
3 = 1
4 = 0
5 = 1
6 = 1

After: [1, 1, 1, 1, 0, 1, 1]
-----
Sintaxe:          (123)10 = (1111011)2
Pressione a tecla ENTER para continuar...
```

Finalidade e funcionalidades

Esse programa teve como finalidade aprender a conversão de bases numéricas, construindo os devidos cálculos e aplicadas a uma linguagem de programação

Fundamentação teórica utilizada

Foi utilizado o método de conversão por multiplicação e divisão e as ferramentas utilizadas foram linguagem de programação Python e visual estúdio code para a compilação do código

Facilidades e dificuldades encontradas Solução implementada

A principal dificuldade desse projeto é a não utilização de funções para conversão direta, nem a utilizações de bibliotecas já prontas do Python, isso tornou o projeto e o problema bem mais complexo de ser resolvido, a solução foi a implementação de mais classes e métodos executando os fundamentos teóricos das conversões por meio de multiplicação e divisão