Vamos começar a programar...

MC102-2018s1-Aula03-180303

Arthur J. Catto, PhD
ajcatto@g.unicamp.br

03 de março de 2018

1 Vamos começar a programar...

Nossa rotina para resolver um problema de programação simples será:

- Entender bem o problema
- Encontrar uma heurística que pareça nos aproximar da solução desejada
- Repetir o passo anterior até que o problema esteja resolvido
- Se necessário, melhorar a solução progressivamente até que o resultado seja satisfatório

1.1 Exemplo 1

Dada uma temperatura em graus Fahrenheit, convertê-la para graus Celsius.

1.1.1 Como resolver este problema?

- Será possível encontrar uma única heurística que resolva o problema?
- Ou será melhor pensar numa sequência de heurísticas?

Vamos experimentar a segunda opção:

```
In [ ]: # Ler a temperatura em graus Fahrenheit
        tempF = float(input("Temperatura em graus Fahrenheit = "))

In [ ]: # Converter essa temperatura para graus Celsius
        tempC = 5 / 9 * (tempF - 32)

In [ ]: # Exibir o resultado
        print("Temperatura em graus Celsius =", tempC)
```

É possível resolver esse problema com um único comando, mas o resultado é bem menos legível...

Em

```
print("Temperatura em graus Celsius =", tempC)
```

substituímos `tempC` por

```
5 / 9 * (tempF - 32)
```

e obtemos

```
print("Temperatura em graus Celsius =", 5 / 9 * (tempF - 32))
```

Depois, substituímos `tempF` por

```
float(input("Temperatura em graus Fahrenheit = "))
```

para obter finalmente

```
print("Temperatura em graus Celsius =",
      5 / 9 * (float(input('Temperatura em graus Fahrenheit = ')) - 32))
```

1.2 Exemplo 2

Um despertador de 24 horas está marcando 13h e está programado para despertar daqui a 50 horas. Quando o alarme disparar, ele estará marcando 15h.

Escreva um programa Python para resolver a versão geral desse problema.

Pergunte ao usuário que horas o despertador está mostrando e, em seguida, o número de horas até o alarme disparar. Seu programa deve calcular que horas o despertador estará marcando quando o alarme tocar.

Como resolver este problema?

```
In [ ]: # Ler a hora atual e o intervalo até o alarme tocar


In [ ]: # Somar a hora atual com o intervalo para obter a hora do alarme


In [ ]: # Exibir o resultado
```

Solução

```
In [ ]: # Ler a hora atual e o intervalo até o alarme tocar
        hora_atual = input("Que horas são? ")
        intervalo = input ("Quantas horas até despertar? ")

        # Somar a hora atual com o intervalo para obter a hora do alarme
        hora_alarme = (hora_atual + intervalo) % 24

        # Exibir o resultado
        print(hora_alarme)
```

1.3 Dicas para depuração de um programa

- Todos são suspeitos até prova em contrário
- Procure pistas
  - Mensagens de erro
  - Comandos de impressão auxiliares
  - Mapas do depurador

1.4 Tire proveito das mensagens de erro

Os autores do livro recomendado registraram os erros cometidos na solução de problemas propostos.

Quase 90% desses erros eram representados por:

- Erros de sintaxe (55%)
- Erros de tipo (14%)
- Erros de nome (11%)
- Erros de valor (10%)

1.4.1 Erros de sintaxe

Ocorrem quando alguma regra de formação da linguagem é violada. São os mais fáceis de encontrar e corrigir por causa das mensagens do interpretador.

Veja se você consegue identificar um ***erro de sintaxe*** no programa abaixo:

```
In [ ]: hora_str = input("Que horas são (0-23)? ")
        espera_str = input("Quantas horas você vai esperar?")

        hora_int = int(hora_str)
        espera_int = int(espera_int)

        alarme_int = hora_int + espera_int
        print(alarme_int)
```

Dicas de pista

- Experimente "comentar" a linha do erro para ver se ele desaparece
- Lembre-se de que, neste caso, o interpretador esperava uma coisa e encontrou outra. Portanto, o erro deve estar do ponto indicado para trás...
- Se comentar a linha "piorar" o erro, tente trocar uma expressão que exista na linha por uma constante razoável e veja se o erro desaparece...

1.4.2 Erros de tipo

Ocorrem quando você tenta combinar objetos incompatíveis e geralmente aparecem em comandos envolvendo expressões matemáticas ou lógicas. Também não costumam ser difíceis de encontrar e corrigir por causa das mensagens do interpretador.

Veja se você consegue identificar um ***erro de tipo*** no programa abaixo:

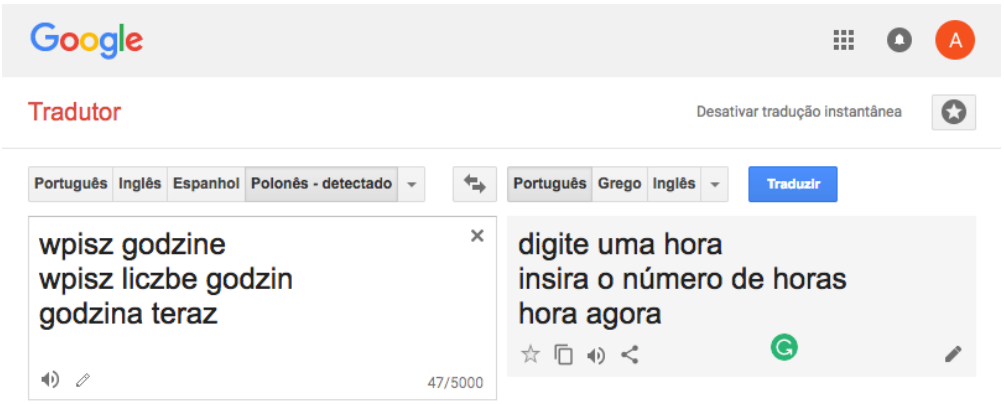
```
In [ ]: a = int(input('wpisz godzine: '))
        x = int(input('wpisz liczbe godzin: '))
        int(x)
        int(a)
        h = x // 24
        s = x % 24
        a = h + s
        print ('godzina teraz =', a)
```

Dicas de pista

- Insira "prints" auxiliares, mostrando o valor e o tipo das variáveis envolvidas na expressão onde você desconfia que o erro deva estar.
- Use nomes adequados para suas variáveis, que o ajudem a lembrar seu tipo e faixa de valores aceitáveis.

Esse programa está correto?

Para responder essa pergunta, primeiro é preciso saber o que se supõe que ele faça...



Depois de corrigir os erros de tipo e traduzir as mensagens, é possível ter uma ideia do que deveria acontecer, mas os nomes das variáveis não ajudam...

```
In [ ]: a = input('digite a hora atual: ')
        x = input('digite o tempo de espera: ')
        x = int(x)
        a = int(a)
        h = x // 24
        s = x % 24
        a = h + s
        print ('hora após a espera:', a)
```

Substituindo textualmente os nomes das variáveis por outros mais razoáveis, obtemos:

```
In [ ]: hora_atual = input('digite a hora atual: ')
        espera = input('digite o tempo de espera: ')
        espera = int(espera)
        hora_atual = int(hora_atual)
        dias_de_espera = espera // 24
        horas_restantes = espera % 24
        hora_atual = dias_de_espera + horas_restantes
        print ('hora após a espera:', hora_atual)
```

E agora deve estar claro para você o que precisa ser corrigido...

1.4.3 Erros de nome

Erros de nome quase sempre indicam que você usou uma variável antes de atribuir um valor a ela e muitas vezes decorrem de simples erros de digitação.

Veja se você consegue identificar um ***erro de nome*** no programa abaixo:

```
In [ ]: que_hora_str = input("Que horas são (0-23)? ")
        espera_str = input("Quantas horas você vai esperar? ")

        que_hora_int = que_hora_str
        print(type(que_hora_int), type(que_hora_str))
        espera_int = int(espera_str)

        alarme_int = que_hora_int + espera_int
        print(alarme_int)
```

Dicas de pista

- Use o editor para buscar o nome no corpo do programa.
- Se ele aparecer apenas uma vez, desconfie...
- Se você "***tiver certeza***" de que ele aparece numa certa linha e o editor não realçá-lo, olhe com mais atenção porque você pode ter cometido um erro de digitação.

Vamos corrigir este programa...

```
In [ ]: n = input("Que horas são? (0-23) ")
        n = imt(n)
        m = input("Daqui a quantas horas vai tocar o alarme? (>= 0) ")
        m = int(m)
        q = m % 12
        print("O despertador vai tocar às", q, "horas.")
```

E mais este...

```
In [ ]: hora_atual = input("Que horas são? (0-23) ")
        delta_alarme = input("Daqui a quantas horas vai tocar o alarme? (>= 0) ")
        int(hora_atual, delta_alarme, hora_alarme)
        hora_alarme = hora_atual + delta_alarme
        print(hora_alarme)
```

1.4.4 Erros de valor

Erros de valor surgem quando você passa um parâmetro para uma função desrespeitando alguma restrição de aceitabilidade. Por exemplo,

```
In [ ]: x = int('abc')
```

Dicas de pista

- Nem sempre um erro de valor decorre de um erro de programação. Veja se o usuário não digitou algum valor incompatível.
- Para evitar isso...
  - Comente seu programa, indicando o propósito e o escopo das variáveis.
  - Use nomes adequados para suas variáveis.
  - Indique os valores aceitáveis nos comandos de entrada.
  - Critique os valores digitados pelo usuário... mas ainda não temos as ferramentas necessárias para isso...

Veja de quantas formas você consegue "quebrar" este programa, digitando valores inaceitáveis:

```
In [ ]: hora_str = input("Que horas são? ")
        hora_int = int(hora_str)

        espera_str = input("Quantas horas você vai esperar? ")
        espera_int = int(espera_str)

        alarme_int = hora_int + espera_int
        print("O alarme vai tocar às", alarme_int)
```

1.5 Exemplo 3

Quantos dias, horas, minutos e segundos há num dado número de segundos?

Como resolver este problema?

```
In [ ]: # Ler o número de segundos a serem convertidos
        s =
        ss = s

In [ ]: # Calcular algumas constantes úteis
        sm =
        sh =
        sd =

In [ ]: # Calcular o número equivalente de dias, horas, minutos e segundos


In [ ]: # Exibir o resultado
        print(          )
```

Solução

```
In [ ]: # Ler o número de segundos a serem convertidos
        segs = int(input('Qual o número de segundos que você quer converter? '))

        # Calcular algumas constantes úteis
        segs_num_minuto = 60
        segs_numa_hora = 60 * segs_num_minuto
        segs_num_dia = 24 * segs_numa_hora

        # Calcular o número equivalente de dias, horas, minutos e segundos
        dias = segs // segs_num_dia
        segs_restantes = segs % segs_no_dia

        horas = segs_restantes // segs_na_hora
        segs_restantes = segs_restantes % segs_na_hora

        minutos = segs_restantes // segs_no_minuto
        segs_restantes = segs_restantes % segs_no_minuto

        # Exibir o resultado
        print(segs, 'segundos =', dias, 'dia(s),', horas, 'hora(s),',
              minutos, 'minuto(s) e', segs_restantes, 'segundo(s).')
```

1.6 Exemplo 4

Quando uma pessoa empresta ou toma emprestado um certo capital, no regime de juros compostos, os juros de cada período são somados ao capital para o cálculo de novos juros nos períodos seguintes. Nesse caso, o valor da dívida é sempre corrigido e a taxa de juros é calculada sobre esse novo valor. A fórmula de juros compostos pode ser escrita como $V_f = V_p \left(1 + \frac{j}{n}\right)^{nt}$, onde

V_f: valor futuro
V_p : valor presente
j : taxa de juros anual nominal
n : número de vezes em que o juro é capitalizado por ano
t : número de anos

Escreva um programa Python que atribua o valor principal de 10000 à variável P, atribua a n o valor 12 e atribua a j uma taxa de juros de 8% (0.08). Depois, peça ao usuário o número de anos, t, ao final do qual ocorrerá o pagamento. Calcule e exiba o valor final após t anos.

```
In [ ]: P =
        n =
        j =

        t =

        F =

        print()
```

Solução

```
In [ ]: P = 10000
        n = 12
        j = 0.08

        t = int(input("Compor juros durante quantos anos? (>= 0) "))

        F = P * (1 + j / n) ** (n * t)
        F = int(100 * F + 0.5) / 100

        print ("O valor do capital após", t, "anos é", F)
```