

Processamento da Informação

Introdução ao curso

Professora Carla Negri Lintzmayer

2019.Q1

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC



- Carla Negri Lintzmayer
- `carla.negri@ufabc.edu.br`
- `carla.negri.ufabc@gmail.com`
- Sala 508-2, bloco A
- `http://professor.ufabc.edu.br/~carla.negri`
- Atendimento: Sextas-feiras, das 10h às 12h

O que vamos aprender neste curso

O que vamos aprender neste curso

Como usar um computador para resolver problemas

1. Definiremos um problema a ser resolvido
2. Discutiremos uma solução para o problema
3. Descreveremos um **algoritmo** para resolver o problema
 - Sequência **bem definida** de comandos e passos, que podem ser especificados de várias formas, inclusive em português
4. Implementaremos este algoritmo e criaremos um **programa**
 - Sequência de comandos e passos que um computador deve executar

Linguagem de programação

- Para construir novos programas, uma forma seria escrever códigos binários diretamente executados por um computador (*hardware*).
- Uma maneira mais simples é escrever os programas em uma linguagem de programação com nível mais alto de abstração.
- Uma **linguagem de programação** é um conjunto de comandos que são mais “próximos” da linguagem humana do que os sinais digitais.
- Nesta disciplina, usaremos a linguagem de programação Java (mas existem muitas outras).

Problema: resolver equação $ax^2 + bx + c = 0$

Algoritmo:

- calcule o discriminante:
 $\Delta = b^2 - 4ac$
- determine a quantidade de raízes: se $\Delta = 0$, existe apenas uma; se $\Delta < 0$, existem duas raízes imaginárias; caso contrário, existem duas raízes reais
- calcule a(s) raiz(es):

$$\frac{-b \pm \sqrt{\Delta}}{2a}$$

Programa em linguagem Java:

```
import java.util.Scanner;
public class EquacaoSegundoGrau {
    public static void main(String[] Strings) {
        Scanner input = new Scanner(System.in);
        double a = input.nextDouble();
        double b = input.nextDouble();
        double c = input.nextDouble();
        double D = b * b - 4 * a * c;
        double r1, r2;

        if (D > 0.0) {
            r1 = (-b + Math.pow(D, 0.5)) / (2.0 * a);
            r2 = (-b - Math.pow(D, 0.5)) / (2.0 * a);
            System.out.println("Raizes: " + r1 + ", " +
                ↵ r2);
        } else if (D == 0.0) {
            r1 = -b / (2.0 * a);
            System.out.println("Raiz: " + r1);
        } else {
            System.out.println("Sem raizes reais.");
        }
    }
}
```

Problema: resolver equação $ax^2 + bx + c = 0$

Algoritmo:

- calcule o discriminante:
 $\Delta = b^2 - 4ac$
- determine a quantidade de raízes: se $\Delta = 0$, existe apenas uma; se $\Delta < 0$, existem duas raízes imaginárias; caso contrário, existem duas raízes reais
- calcule a(s) raiz(es):
$$\frac{-b \pm \sqrt{\Delta}}{2a}$$

Programa em linguagem C:

```
#include <stdio.h>
#include <math.h>

int main() {
    double a, b, c;
    scanf("%lf %lf %lf", &a, &b, &c);
    double D = b * b - 4 * a * c;
    double r1, r2;

    if (D > 0.0) {
        r1 = (-b + sqrt(D)) / (2.0 * a);
        r2 = (-b - sqrt(D)) / (2.0 * a);
        printf("Raizes: %lf, %lf\n", r1, r2);
    } else if (D == 0.0) {
        r1 = -b / (2.0 * a);
        printf("Raiz: %lf\n", r1);
    } else {
        printf("Não há raízes reais.\n");
    }
}
```

Problema: resolver equação $ax^2 + bx + c = 0$

Algoritmo:

- calcule o discriminante:

$$\Delta = b^2 - 4ac$$

- determine a quantidade de raízes: se $\Delta = 0$, existe apenas uma; se $\Delta < 0$, existem duas raízes imaginárias; caso contrário, existem duas raízes reais

- calcule a(s) raiz(es):

$$\frac{-b \pm \sqrt{\Delta}}{2a}$$

Programa em linguagem Python:

```
from math import sqrt

a = float(input())
b = float(input())
c = float(input())
D = b**2 - 4*a*c

if D > 0:
    r1 = (((-b) + sqrt(D))/(2*a))
    r2 = (((-b) - sqrt(D))/(2*a))
    print("Raizes: %f, %f" % (r1, r2))
elif D == 0:
    r = (-b) / 2*a
    print("Raiz: ", x)
else:
    print("Não há raízes reais.")
```

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

- Objetivo: **praticar** aspectos importantes de lógica de programação
- Breve revisão da parte teórica
- Resolução de alguns exercícios **em conjunto**
- Atendimento de dúvidas na resolução de outros **exercícios**
- <http://professor.ufabc.edu.br/~carla.negri>

O que eu espero de você

- **Participe das aulas**
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que eu espero de você

- Participe das aulas
- Respeite os horários
- Não deixe dúvidas acumularem
- Faça e implemente os exercícios
- Comporte-se de acordo com o código de ética da UFABC
 - Seja autor das suas soluções
 - Não assine a lista de presença por outros
 - Não consulte nada durante as avaliações

O que você pode esperar de mim



O que vocês podem esperar de mim

- Participação
- Paciência
- Dedicção
- Atendimento

- Nessa disciplina aplicaremos os conceitos vistos em linguagem Java
- Usaremos um editor de texto simples para escrever os códigos (Windows: Notepad++; Linux: Gedit, Kate, Gvim, Emacs, ...)
- O código fonte será transformado em programa por meio de um terminal
- Os exercícios deverão ser submetidos no URI
 - <https://www.urionlinejudge.com.br/judge/pt/login>