

Programação Orientada a Objetos

Classe, objeto, atributos, métodos,
construtor

Classes e Objetos

OBJETOS

- Computacionalmente, são entidades identificáveis que representam algo real ou imaginário, podendo ser concreto ou abstrato.
- São variáveis cujo tipo é uma classe bem definida

EXEMPLOS: o celular S22 da samsung, o fusca da Volkswagen, o horário 13h20m30s, o quadro monalisa, o cadastro do Emanuel, ...

- Não necessita ser real ou palpável
- Possuem características que o definem: **atributos** (**variáveis da sua classe**)
- Possuem comportamentos/ operações : **métodos** (**funções da sua classe**)
- Definem instâncias da classe na qual eles pertencem

Classes e Objetos

CLASSES

- Objetos de um mesmo tipo possuem o mesmo comportamento
- Objetos de um mesmo tipo pertencem à mesma classe
- A classe representa a abstração de seus objetos

EXEMPLOS: Celular, Carro, Horário, Quadro, Cadastro, ...

ESTRUTURAÇÃO:

- Atributos : variáveis da classe
- Métodos : funções da classe
- Construtor : método especial da classe usado para inicializar os atributos de um objeto recém-criado
- O construtor de um classe não possui tipo de retorno e deve conter o mesmo nome da classe à qual ele pertence

Cachorro

raca : String
sexo : char
idade : int
cor : String

Cachorro(String, char, int, String)

getRaca() : String
getSexo() : char
getIdade() : int
getCor() : String

setRaca(String) : void
setSexo(char) : void
setIdade(int) : void
setCor(String) : void

rex

“pastor alemão” ,
m, 12,
“preto acinzentado”

lili

“poodle” ,
f, 03,
“branca”

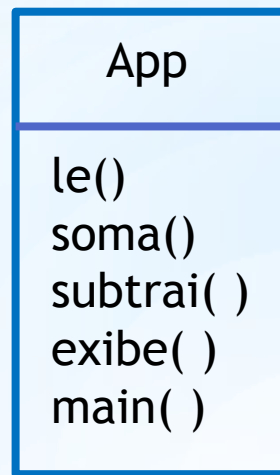
nina

“pinsher” ,
f, 26,
“marrom”

Padronização na POO

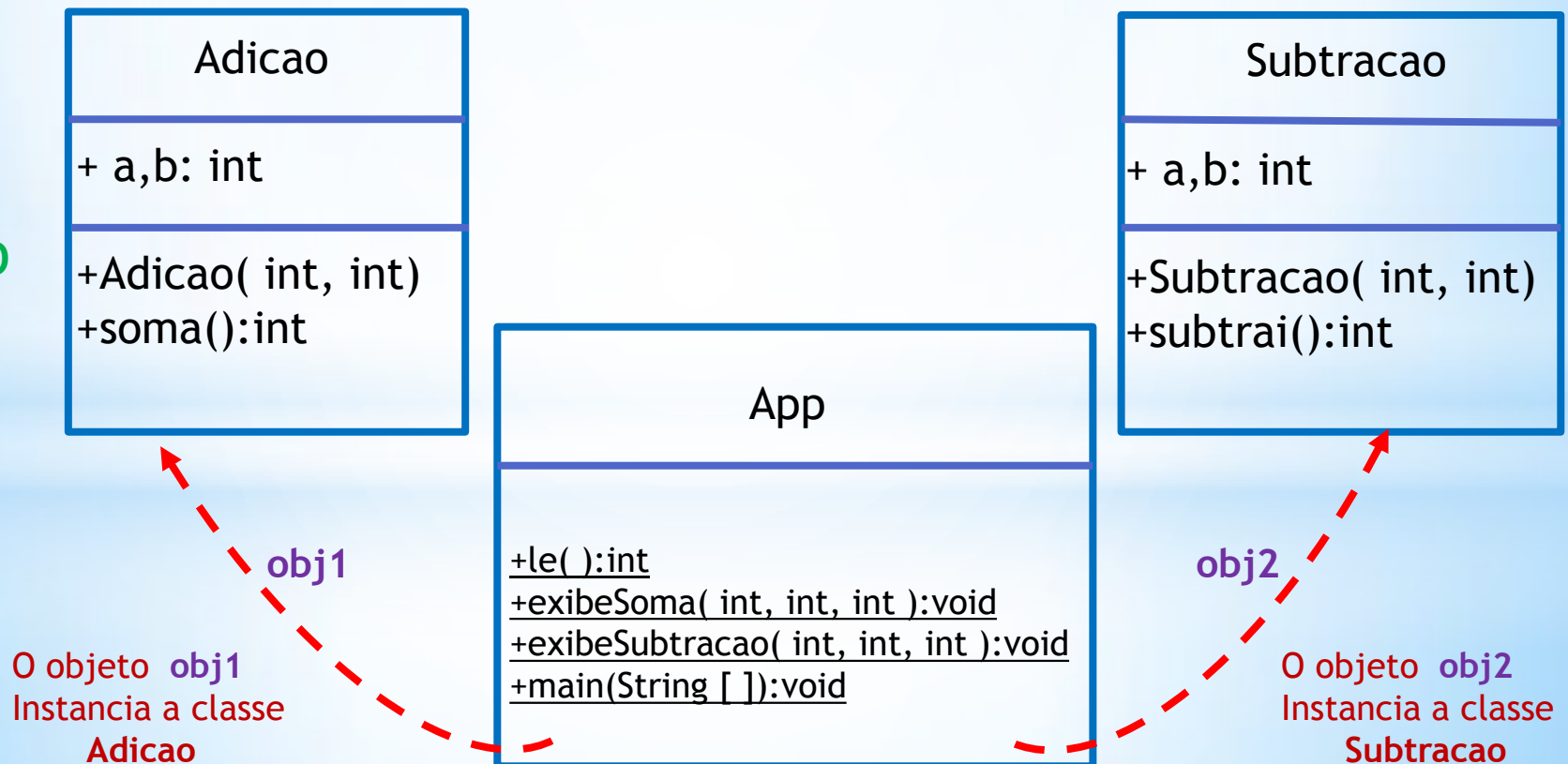
- Todo nome de classe inicia com letra maiúscula
- Variáveis e métodos inicia com letra minúsculas
- Nomes compostos iniciam as palavras intermediárias com maiúsculas
- Constantes possuem todas as letras maiúsculas

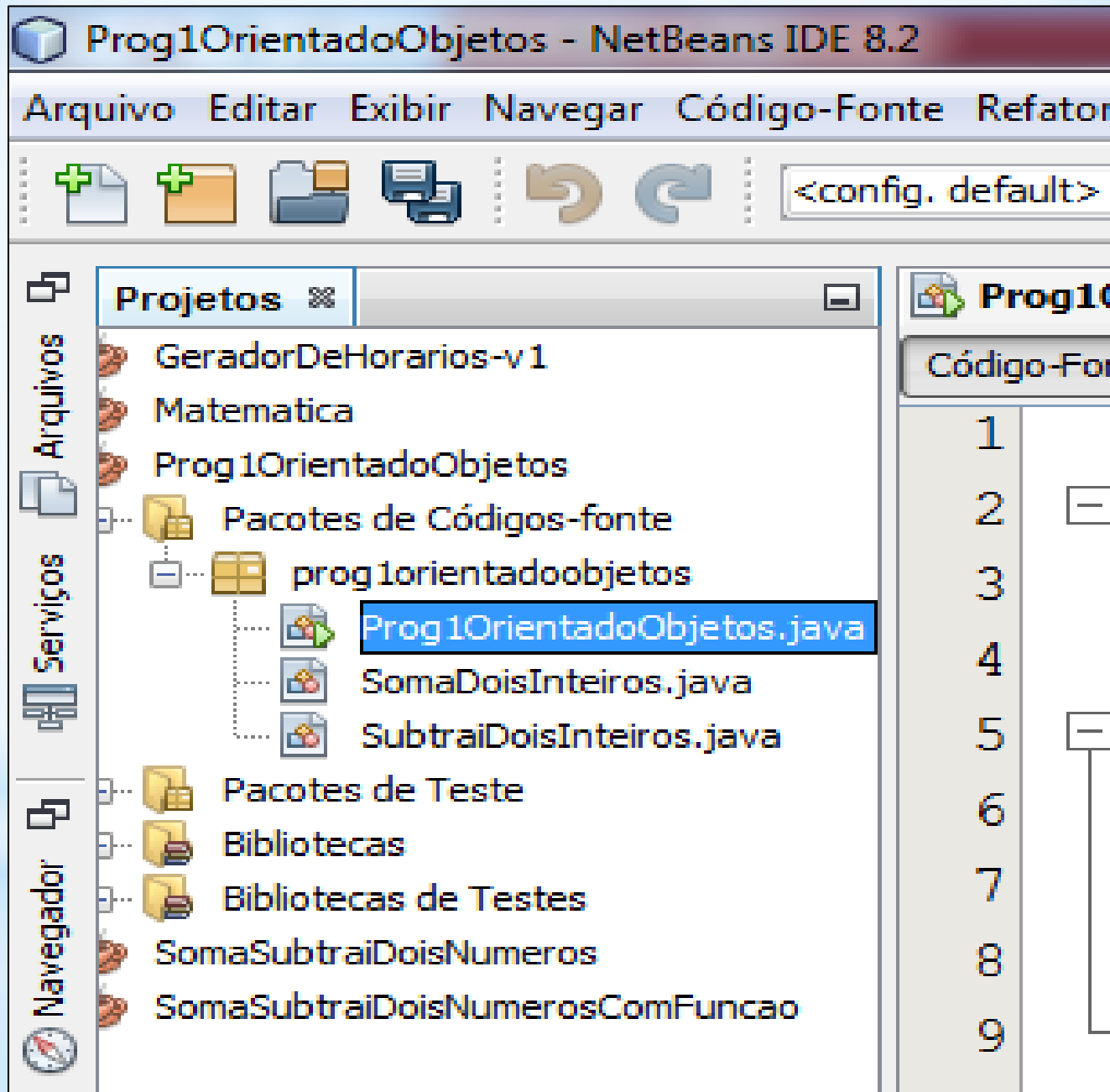
sem POO

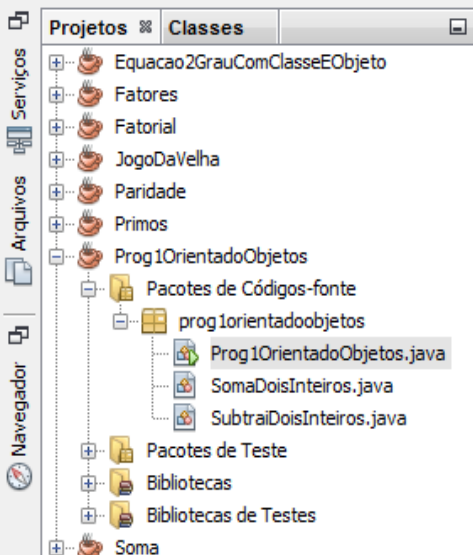


1 classe contendo a função main()
e todas as outras funções

com POO

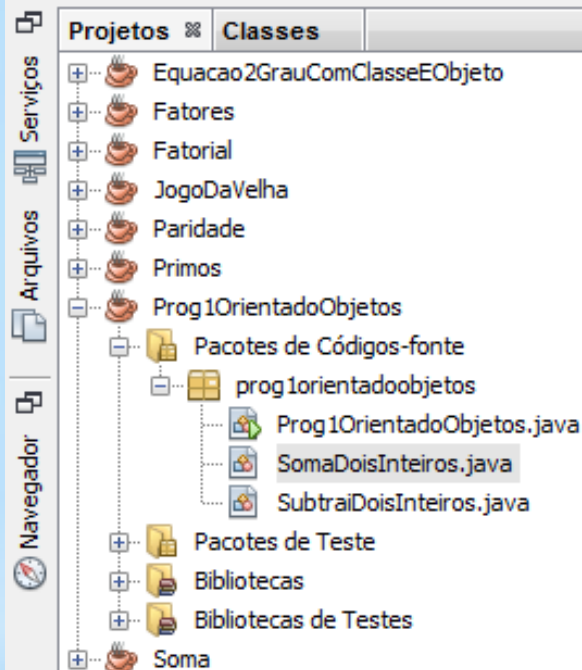
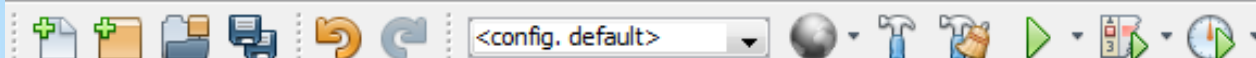






```

5  public class Prog1OrientadoObjetos {
6
7  +   public static int le() {...5 linhas }
12
13 +   public static void exibeSoma(int a, int b, int s) {...3 linhas }
16
17 +   public static void exibeSubtracao(int a, int b, int s) {...3 linhas }
20
21 -   public static void main(String[] args) {
22       int a, b, s;
23       a = le();
24       b = le();
25       SomaDoisInteiros adicao = new SomaDoisInteiros(a,b);
26       s = adicao.soma();
27       exibeSoma(a,b,s);
28       SubtraiDoisInteiros subtracao = new SubtraiDoisInteiros(a,b);
29       s = subtracao.subtrai();
30       exibeSubtracao(a,b,s);
31   }
32
33 }
```

Prog1OrientadoObjetos.java SomaDoisInteiros.java SubtraiDoisInteiros.java

Código-Fonte

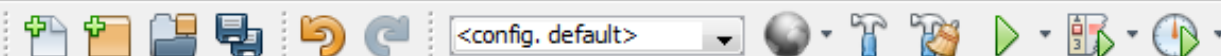
Histórico



```

1 package proglorientadoobjetos;
2
3 public class SomaDoisInteiros {
4     public int a, b;
5
6     public SomaDoisInteiros(int x, int y) {
7         a = x;
8         b = y;
9     }
10
11     public int soma() {
12         return a+b;
13     }
14
15 }
16

```



Projetos **Classes**

- Equacao2GrauComClasseEObjeto
- Fatores
- Fatorial
- JogoDaVelha
- Paridade
- Primos
- Prog1OrientadoObjetos
 - Pacotes de Códigos-fonte
 - prog1orientadoobjetos
 - Prog1OrientadoObjetos.java
 - SomaDoisInteiros.java
 - SubtraiDoisInteiros.java
 - Pacotes de Teste
 - Bibliotecas
 - Bibliotecas de Testes
 - Soma

Prog1OrientadoObjetos.java SomaDoisInteiros.java SubtraiDoisInteiros.java

Código-Fonte

Histórico

```

1  package proglorientadoobjetos;
2
3  public class SubtraiDoisInteiros {
4      public int a, b;
5
6      public SubtraiDoisInteiros(int x, int y) {
7          a = x;
8          b = y;
9      }
10
11     public int subtrai() {
12         return a-b;
13     }
14
15 }
16

```