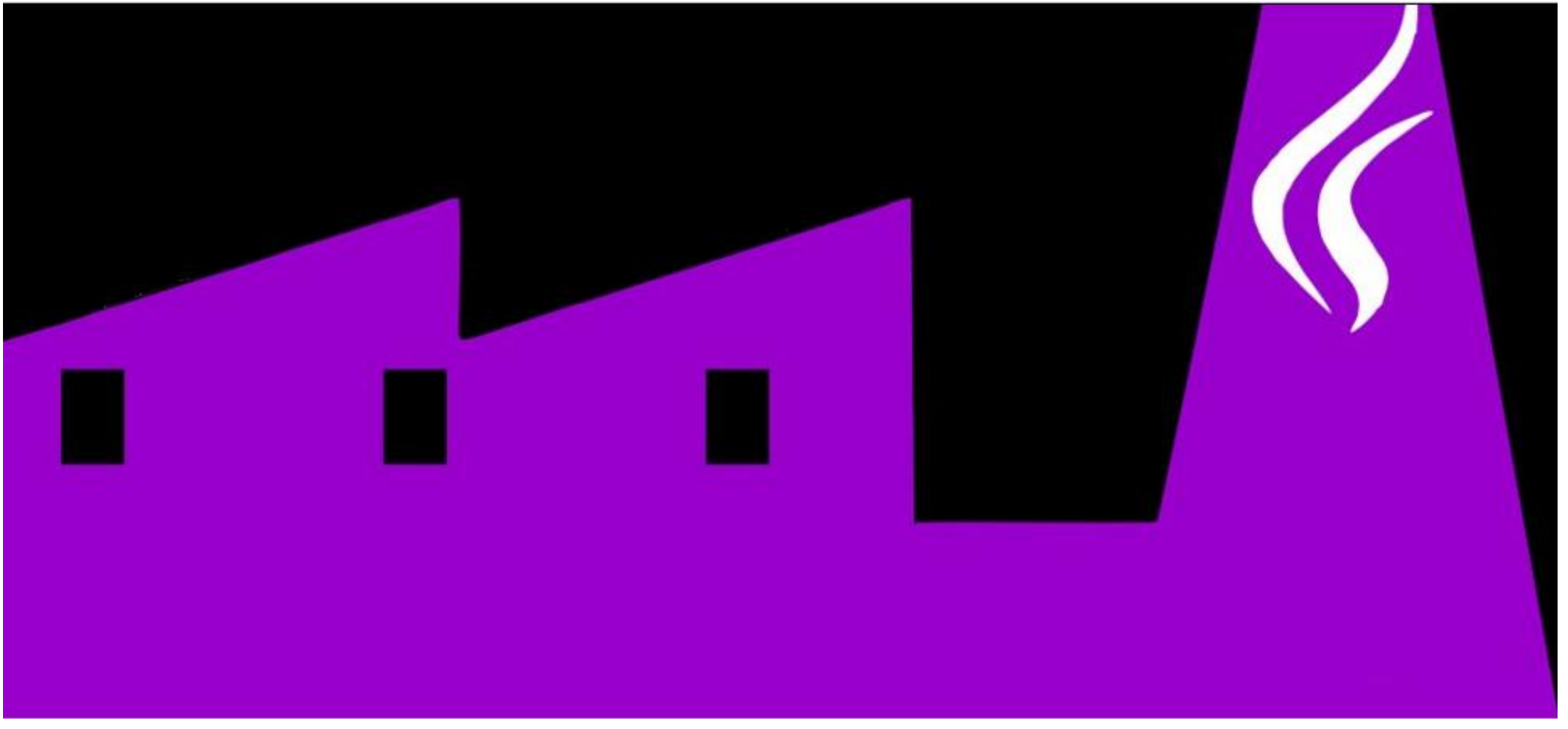
Fábrica de Software

Prof[es.] Ivan L. Süptitz e Evandro Franzen

Atributos e métodos de classe (static);

Sobrecarga de métodos e múltiplos construtores



Variáveis estáticas versus atributos

- Enquanto cada instância da classe (objeto) tem seus próprios atributos...
- Variáveis estáticas (ou de classe) são compartilhadas por todas as instâncias da classe

Métodos estáticos

Da mesma forma que há variáveis estáticas (de classe) e variáveis de instância (atributos), há métodos estáticos (de classe) e métodos de instância.

Importante: Mesmo que um método estático esteja presente na mesma classe dos atributos, um método estático não tem acesso a estes atributos.

Métodos estáticos

```
class Calcula{
    public static int maior (int x, int y) {
        if (x > y) return x;
        else return y;
    }

    public static void main (String [ ] args) {
        int m, x, y;

        x = new Scanner(System.in).nextInt();
        y = new Scanner(System.in).nextInt();

        m = maior (x, y);

        System.out.println(m);
    }
}
```

Métodos estáticos

- Para chamar um método estático a partir de outra classe, chamamos colocando o nome da classe, seguido de ponto e o nome do método. Ex:

```
Calcula.maior(3,2);
```

- Não precisa instanciar objeto.

- Exemplo comum é a Classe **Math** que tem muitas funções matemáticas, todas como métodos estáticos.

CUIDADO: usar só métodos estáticos não é OO!!

Exercícios Métodos Estáticos

J.4.1 A distância média da Terra à Lua é de aproximadamente 382.000 quilômetros. Usando uma classe com métodos estáticos escreva um programa que mostre qual é a distância média da Terra à Lua em milhas e pés.

J.4.2 Escreva a classe ConversaoDeUnidadesDeArea com métodos estáticos para conversão das unidades de área segundo a lista abaixo.

- 1 metro quadrado = 10.76 pés quadrados
- 1 pé quadrado = 929 centímetros quadrados
- 1 milha quadrada = 640 acres
- 1 acre = 43.560 pés quadrados

J.4.3 A área de um campo de futebol é de 8.250 metros quadrados. Usando a classe ConversaoDeUnidadesDeArea, exercício 2, escreva um programa em Java que mostre qual é a área de um campo de futebol em pés quadrados, acres e centímetros quadrados. Escreva métodos adicionais para a classe ConversaoDeUnidadesDeArea, se necessário.

Exercícios Métodos Estáticos

J.4.4 Escreva a classe ConversaoDeUnidadesDeTempo com métodos estáticos para conversão aproximada das unidades de tempo segundo a lista abaixo.

- 1 minuto = 60 segundos
- 1 hora = 60 minutos
- 1 dia = 24 horas
- 1 semana = 7 dias
- 1 mês = 30 dias
- 1 ano = 365.25 dias

J.4.5 O tempo de gestação de um elefante indiano é de aproximadamente 624 dias. Usando a classe ConversaoDeUnidadesDeTempo, exercício anterior, escreva um programa que mostre qual é o tempo de gestação de um elefante indiano em dias, horas, minutos e segundos. Escreva métodos adicionais para a classe ConversaoDeUnidadesDeTempo, se necessário.

Exercícios Métodos Estáticos

J.4.6 Escreva uma classe que contenha métodos estáticos para calcular as médias e somas de dois, três, quatro e cinco valores, considerando que deve haver métodos para os tipos int e double. No total, 16 métodos deverão ser criados.

Sobrecarga de Métodos

A sobrecarga de métodos, também conhecida como *overloading*, ocorre quando criamos dois ou mais métodos com o mesmo nome mas com uma lista de argumentos diferente.

A sobrecarga está sendo aplicada, por exemplo, quando são declarados dois construtores para uma mesma classe, sendo um sem parâmetros e outro com parâmetros.

Sobrecarga de Métodos

```
public class Conta {

    private double saldo;

    public Conta () {
    }

    public Conta (double saldo) {
        this.saldo = saldo;
    }
}
```

Podemos, então, criar objetos dessa classe de dois modos:

Conta c = new Conta(); ou

Conta c = new Conta(100); ⇒ inicializando o saldo com algum valor

Sobrecarga de Métodos

- também é aplicada para os outros métodos que não sejam construtores.

Em algumas situações, é útil poder executar um método em uma classe passando mais ou menos argumentos, conforme a necessidade.

Também pode ser necessário fazer a sobrecarga de um mesmo método, com o mesmo número de argumentos, porém variando os tipos desses argumentos.

Sobrecarga de Métodos

```
public class Conta {
    private double saldo;
    public String toString() {
        return " saldo = " + this.saldo;
    }
    public String toString(String prefixo) {
        return prefixo + ": " + toString();
    }
}
```

Exemplo de uso:

```
Conta c = new Conta(100);
c.toString();   ou
c.toString("Conta número X");
```

Sobrecarga de Métodos

- Para permitir a sobrecarga, os nomes dos métodos devem ser iguais mas as assinaturas devem ser diferentes.
- Duas assinaturas idênticas não são permitidas.

Sobrecarga de Métodos

- A assinatura de um método é composta por seu nome e pelo número e tipos de argumentos que são passados para esse método, independentemente dos nomes das variáveis usadas na declaração do método.

Sobrecarga de Métodos

```
public class Data {

   private int dia;
   private int mes;
   private int ano;

   public Data(int dia, int mes, int ano) {
      this.dia = dia;
      this.mes = mes;
      this.ano = ano;
   }

   public Data(int mes, int dia, int ano) {
      this.dia = dia;
      this.mes = mes;
      this.ano = ano;
   }
}
```

Como todos os argumentos são do mesmo tipo, somente a inversão do nome das variáveis não é suficiente para que esses dois construtores sejam considerados diferentes.

Esse código, portanto, geraria um erro de compilação "método duplicado na classe Data".

Sobrecarga de Métodos

- O tipo de retorno do método não é considerado parte da assinatura.

Sendo assim, não podemos ter dois métodos com o mesmo nome e com o mesmo número e tipos de argumentos, e apenas com o tipo de retorno diferente.

Sobrecarga de Métodos

Na classe Data declarada anteriormente não seria possível a declaração dos dois métodos abaixo, pois suas assinaturas são idênticas, eles só diferem quanto ao tipo de retorno, que não é considerado parte da assinatura do método:

```
public String getData( ) {
   return toString( );
}
```

```
public Date getData( ) {
    Date d = new Date( );
    d.setDate(dia);
    d.setMonth(mes);
    d.setYear(ano);
    return d;
}
```

Sobrecarga de Métodos

A decisão sobre qual método será chamado quando existirem dois ou mais métodos com o mesmo nome será feita pelo compilador, que verificará se os tipos passados como argumentos casam com alguma das assinaturas daquele método.

Supondo que os argumentos sejam de tipos diferentes, a ordem em que aparecem também é relevante, isto é:

- `public Data(int dia, String mes, long ano)` ≠
- `public Data(String mes, int dia, long ano)`

Exercícios Sobrecarga

J.4.7 Refazer a questão **J.4.6**. Ainda deve haver 16 métodos no total, mas os nomes só podem ser “soma” e “media”

J.4.8 Criar no projeto da caneta três métodos construtores com sobrecarga. Considerar os construtores diferentes descritos abaixo:

a. Construtor sem parâmetros, cria uma caneta tampada e da cor azul;
b. Construtor que recebe a cor e o modelo como parâmetro e, também estará tampada;
c. Construtor que recebe como parâmetro, cor, ponta e modelo (também estará tampada).

J.4.9 No projeto da caneta, criar o método modificarAtributos(). Sendo que:

a. Recebe como parâmetros, a cor;
b. Recebe como parâmetros a cor, o modelo;
c. Recebe como parâmetros a cor, o modelo e a ponta.