

Programa Principal

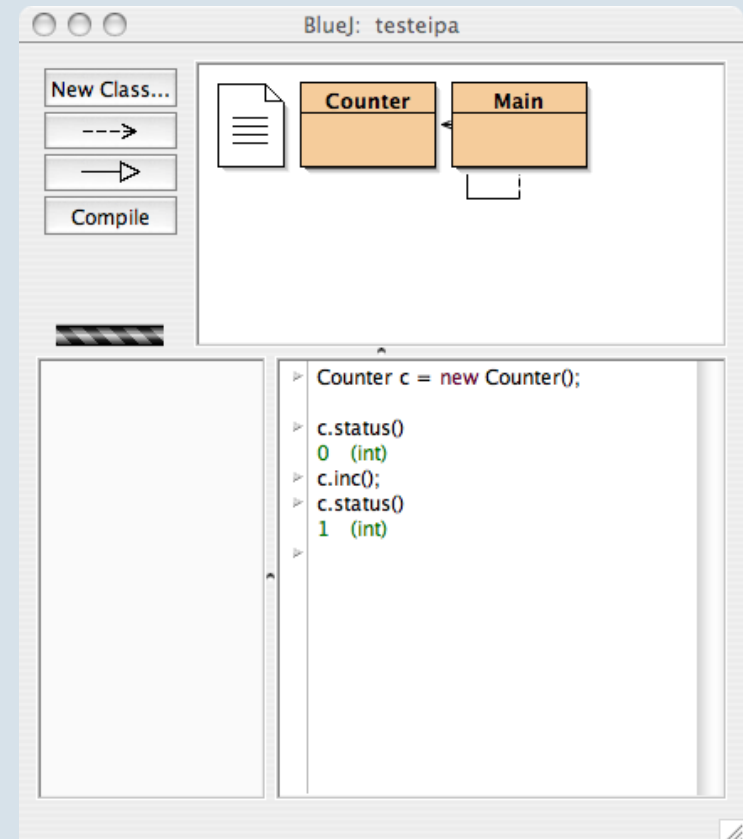
Leitura e Escrita de dados

Fernanda Barbosa
Fernando Birra
Luís Caires
Armanda Rodrigues

Programa Principal Simple

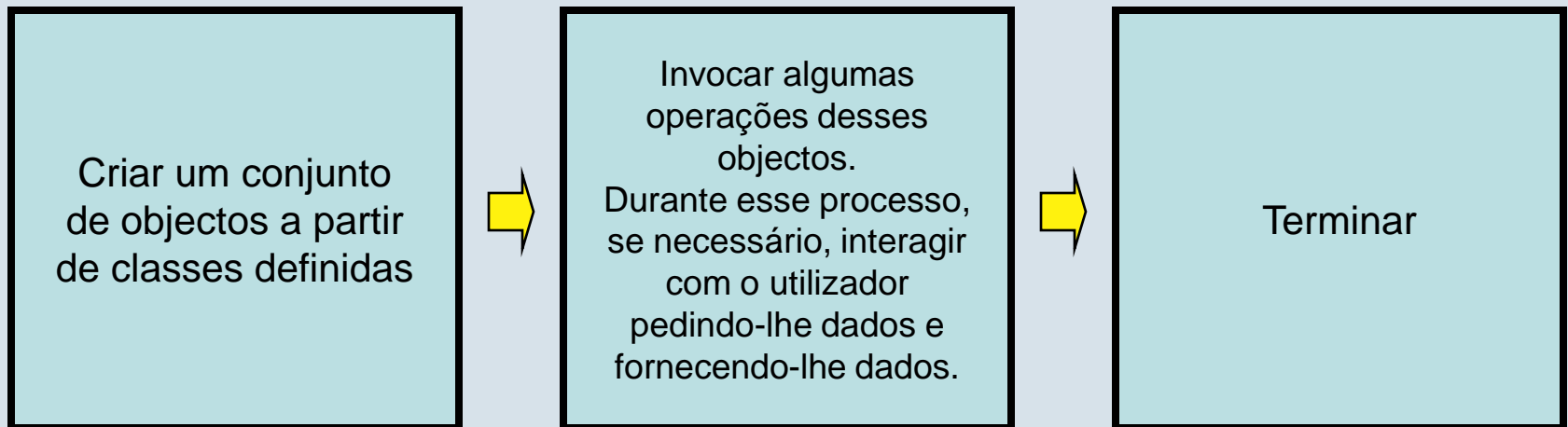
Programa Principal Simples

- Já vimos como programar classes de objectos em Java, como criar objectos e como interagir com objectos usando o “code pad” do IDE BlueJ
- No entanto, a programação visa em geral a construção de software que pode ser executado **autonomamente**, sem necessitar do IDE

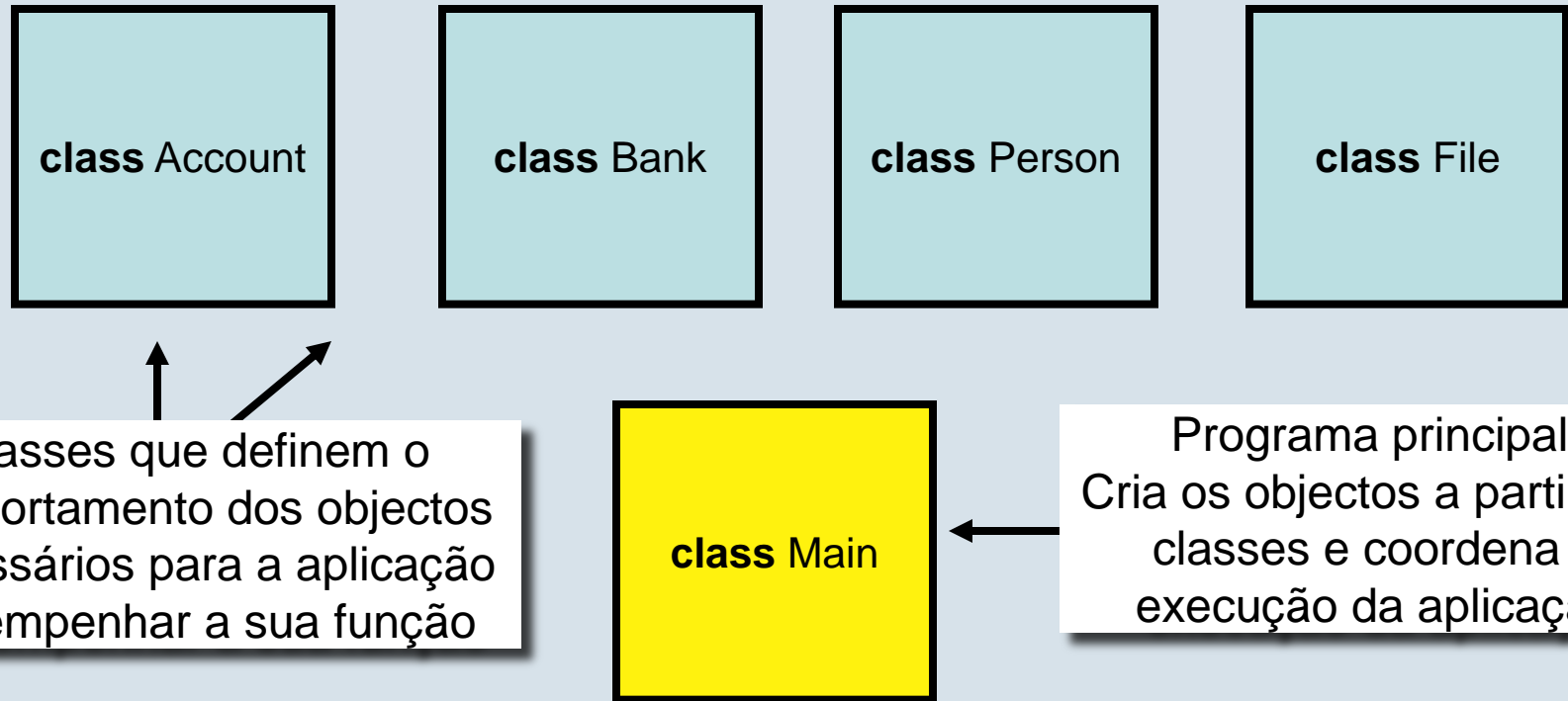


Programa Principal Simples

- Os programas mais simples têm sempre a seguinte estrutura de tarefas em sequência



Estrutura de uma Aplicação Simples



- Os componentes de qualquer aplicação são sempre **um conjunto de classes** e um **programa principal**, que coordena a execução do sistema de software

Programa Principal Simples

- Em Java, o programa principal de uma aplicação define-se numa classe especial (a que vamos chamar `Main`), definida à parte das outras classes
- Na classe `Main` define-se um método especial, estático, chamado `main`, com o programa principal

```
public class Main {  
    public static void main(String[] args) {
```

Programa principal

Quando a aplicação iniciar a sua execução, são as instruções aqui colocadas que irão ser executadas

```
}
```

```
}
```

Programa Principal Simples

- Exemplo (Lâmpadas)
 - O programa principal de uma aplicação simples que cria duas lâmpadas e realiza algumas operações

```
public class Main {  
    public static void main(String[] args) {  
        Lamp l1 = new Lamp();  
        Lamp l2 = new Lamp();  
        l1.on();  
        l2.on();  
        l1.off();  
    }  
}
```

Programa Principal Simples

- Exemplo (Conta)
 - O programa principal de uma aplicação simples que cria uma conta bancária segura e realiza algumas operações

```
public class Main {  
    public static void main(String[] args) {  
        SafeBankAccount b = new SafeBankAccount(1000);  
  
        b.deposit(20);  
        b.applyInterest();  
    }  
}
```


Programa Principal Simples

- Exemplo (Contas)

- O programa principal de uma aplicação simples que cria duas contas e realiza uma transferência entre elas

```
public class Main {  
    public static void main(String[] args) {  
        SafeBankAccount b1 = new SafeBankAccount (1000) ;  
        SafeBankAccount b2 = new SafeBankAccount (2000) ;  
        int some_amount = 20 ;  
        b1.withdraw (some_amount) ;  
        b2.deposit (some_amount) ;  
    }  
}
```

Output de texto

- Exemplo (Output)

A operação `System.out.println` pode ser usada no programa principal para escrever mensagens ao utilizador

```
public class Main {  
    public static void main(String[] args) {  
        String name = "Luis";  
        int age = 20;  
  
        System.out.println(name);  
        System.out.println("The age is "+age+" , not 42");  
  
        System.out.println("Pi="+Math.PI);  
    }  
}
```

Programa Principal Simples

- Exemplo (Contas e Output)
 - O programa principal de uma aplicação simples que cria duas contas e realiza uma transferência entre elas

```
public class Main {  
    public static void main(String[] args) {  
        SafeBankAccount b1 = new SafeBankAccount (1000);  
        SafeBankAccount b2 = new SafeBankAccount (2000);  
        int some_amount = 20;  
        b1.withdraw(some_amount);  
        b2.deposit(some_amount);  
        System.out.println( b1.getBalance() );  
        System.out.println( b2.getBalance() );  
    }  
}
```

Disciplina de Uso

- Em qualquer aplicação que se programe é muito importante separar claramente
 - As partes responsáveis pela interacção com o utilizador
 - As partes responsáveis pela funcionalidade dos vários objectos (definidas em várias classes)
- Assim, o uso de operações de output (ou input, como veremos mais tarde) é apenas permitido no interior da classe `Main`. Esta regra é muito importante.
- Por exemplo, o uso de `System.out.println` numa classe de aplicação, como `BankAccount` ou `Calculator` será **sempre proibido** nas disciplinas de programação da LEI.

Recorde a Conta Bancária Segura

- Operações (interface de SafeBankAccount):

```
public void deposit(int amount)
```

Depositar a importância amount na conta

```
public void withdraw(int amount)
```

Levantar a importância amount na conta ou aplicar a multa

```
public int getBalance()
```

Consultar o saldo da conta

```
public boolean redZone()
```

Indica se a conta está devedora

```
public int computeInterest()
```

Calcular qual o valor do juro anual a aplicar

```
public void applyInterest()
```

Creditar o juro anual ao saldo da conta

Exercício

- Desenvolver um programa que:
 - Pede ao utilizador para criar uma conta bancária com um saldo inicial.
 - Pede o valor a levantar na conta. E informa, caso não seja possível efectuar o levantamento.
 - O levantamento só será efectuado com sucesso, se o saldo da conta for igual ou superior ao valor a levantar;
 - Note que, caso não seja efectuado o levantamento será sempre aplicada uma multa (recorde a definição de conta segura).
 - Lista o saldo corrente da conta.

Leitura de dados

- `System.in` é o nome de um objecto pré-definido utilizado para fazer leitura de dados (o teclado)
- No entanto, o `System.in` só lê um caracter de cada vez
- No Java, a classe `Scanner` foi criada para fazer a leitura a partir do teclado de uma forma simples

```
int value;  
Scanner in = new Scanner(System.in);  
System.out.print("Quantidade: ");  
value = in.nextInt();
```

- `nextInt()` lê um inteiro
- `nextFloat()` lê um float
- `nextDouble()` lê um double
- `next()` lê uma string sem espaços
- `nextLine()` lê uma linha (até ao Enter)

Estude a classe `Scanner` no pacote `java.util`

Main.java

```
import java.util.Scanner;
public class Main {
```

Temos que indicar onde está definida a classe Scanner

```
    public static void main(String[] args) {
```

```
        BankAccount account;
```

```
        int value;
```

```
        Scanner in = new Scanner(System.in);
```

```
        System.out.println("Saldo inicial da conta:");
```

```
        value = in.nextInt();
```

```
        in.nextLine();
```

Temos que ler tudo o que está para além do inteiro

```
        account = new BankAccount(value);
```

```
        System.out.println("Valor a levantar:");
```

```
        value = in.nextInt();
```

```
        in.nextLine();
```

```
        if (account.getBalance() < value)
```

```
            System.out.println("Levantamento não efectuado");
```

```
        else {
```

```
            account.withdraw(value);
```

```
            System.out.println("Saldo da conta: "
```

```
                +account.getBalance()+" Centimos");
```

```
        }
```

```
    }
```

```
}
```


Exercício

- Desenvolver um programa que:
 - Pede ao utilizador, a informação referente a duas contas bancárias.
 - Para cada conta pede o saldo inicial da conta
 - Pede o valor a transferir da primeira conta dada para a segunda conta e efectua a transferência. E informa se a transferência foi ou não efectuada com sucesso.
 - O processo de transferência consiste em levantar dinheiro na primeira conta e depositar esse mesmo dinheiro na segunda conta;
 - Uma transferência só será efectuada com sucesso, se for possível efectuar o levantamento na primeira conta.
 - Neste caso, se o saldo for inferior ao valor a transferir, não há lugar a débito de multa.
 - Lista a informação referente a cada conta:
 - Para cada conta deve escrever o saldo da conta.

Main.java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        int amount;
        BankAccount accountO;
        BankAccount accountD;
        Scanner in = new Scanner(System.in);
        System.out.println("Conta Origem");
        System.out.println("Saldo inicial em centimos:");
        amount = in.nextInt();
        in.nextLine();
        accountO = new BankAccount(amount);
        System.out.println("Conta Destino");
        System.out.println("Saldo inicial em centimos:");
        amount = in.nextInt();
        in.nextLine();
        accountD = new BankAccount(amount);
        ...
    }
}
```

**Cria as duas contas,
com os saldos
iniciais dados pelo
utilizador**

Ficheiro Main.java

```
public class Main {
```

Pede o valor a transferir e efectua transferência

```
    public static void main(String[] args) {
        int value;
        ...
        System.out.println("Valor a transferir em centimos:");
        value = in.nextInt();
        in.nextLine();
        if (accountO.getBalance() >= value) {
            accountO.withdraw(value);
            accountD.deposit(value);
            System.out.println("Transferencia efectuada com sucesso");
        }
        else
            System.out.println("Transferencia efectuada sem sucesso");
        System.out.println("Conta Origem");
        System.out.println(accountO.getBalance() + " Centimos");
        System.out.println("Conta Destino");
        System.out.println(accountD.getBalance() + " Centimos");
    }
}
```

Recorde a Estação Meteorológica

- Cada objecto `WeatherStation` recebe valores de temperatura ao longo do tempo e guarda o máximo, o mínimo e a média das temperaturas registadas
- Operações reconhecidas

```
public void sampleTemperature(double temp)
```

Registar a amostra `temp` na estação

```
public double getMaximum()
```

Consultar a máxima temperatura observada até ao momento

```
public double getMinimum()
```

Consultar a mínima temperatura observada até ao momento

```
public double getAverage()
```

Consultar a média das temperaturas observadas até ao momento

```
public void reset(double temp)
```

Apagar a informação registada na estação, para se iniciarem novas medições. É indicada também uma primeira nova medição.

Exercício

- Desenvolver um programa que:
 - Pede ao utilizador, a informação referente a 5 temperaturas registadas numa dada localidade.
 - As temperaturas devem ser dadas todas na mesma linha separadas por espaço.
 - Lista a informação referente a:
 - Temperatura máxima registada;
 - Temperatura mínima registada;
 - Média das temperaturas registadas.

Main.java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        double t;
        WeatherStation local;
        Scanner in = new Scanner(System.in);
        System.out.println("Introduzir 5 temperaturas");
        t = in.nextDouble();
        local = new WeatherStation(t);
        t = in.nextDouble();
        local.sampleTemperature(t);
        t = in.nextDouble();
        local.sampleTemperature(t);
        t = in.nextDouble();
        local.sampleTemperature(t);
        t = in.nextDouble();
        local.sampleTemperature(t);
        in.nextLine();
        ...
    }
}
```

**Leitura e registo das
cinco temperaturas**

Main.java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        ...
        System.out.println("Maxima "+ local.getMaximum());
        System.out.println("Minima "+ local.getMinimum());
        System.out.println("Media "+ local.getAverage());
    }
}
```

**Escrever as temperaturas
máxima, mínima e média**

Disciplina de Uso

- Em qualquer aplicação que se programe é muito importante separar claramente
 - As partes responsáveis pela interação com o utilizador
 - As partes responsáveis pela funcionalidade dos vários objectos (definidas em várias classes)
- Assim, **o uso de operações de output e input é apenas permitido no interior da classe Main.**
- Esta regra é **mesmo** muito importante.