INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Pacotes e Jar

Pacotes

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Usamos **pacotes** em **java** para **organizar** as **classes semelhantes**.

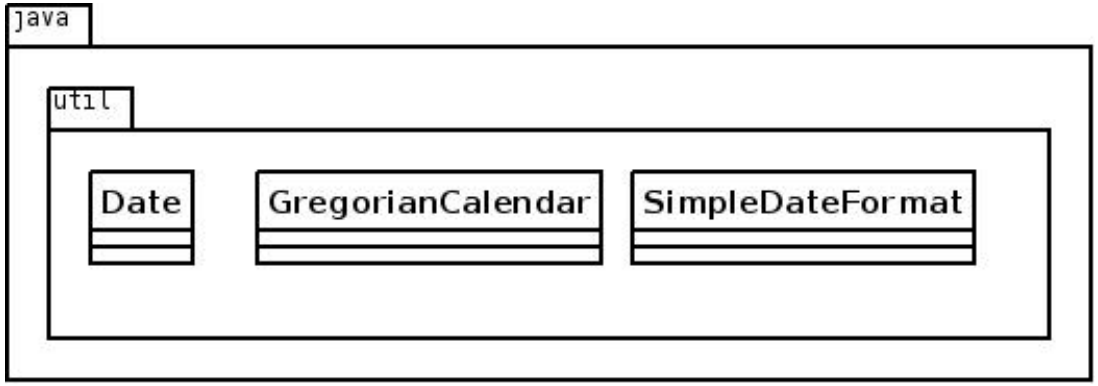
- **Pacotes**, a grosso modo, são **apenas pastas** ou **diretórios** do **sistema operacional** onde ficam **armazenados** os **arquivos** fonte de **Java**.

- Os **pacotes** são **essenciais** para o **conceito** de **encapsulamento**, no qual são **dados níveis** de **acesso** as classes.

Pacotes

- Os **diretórios** estão **relacionados** aos chamados **pacotes** e costumam **agrupar classes** de **funcionalidades similares** ou **relacionadas**.

- Por exemplo, no pacote **java.util** temos as classes **Date**, **SimpleDateFormat** e **GregorianCalendar**; todas elas **trabalham** com **datas** de **formas diferentes.**



Pacotes

- Se a classe **Cliente** está no **pacote banco**, ela deverá estar no **diretório** com o mesmo **nome**: **banco**. Se ela se localiza no **pacote br.com.ifba.banco**, **significa** que está no **diretório br/com/ifba/banco**.

- A classe **Cliente**, que se **localiza** nesse **último diretório**, deve ser escrita da seguinte forma:

```
package br.com.ifba.banco;
class Cliente { // ... }
```

- Fica fácil **notar** que a **palavra chave** **package** indica qual o **pacote/diretório** contém esta classe.

- Um **pacote** pode conter **nenhum** ou mais **subpacotes** e/ou **classes** dentro dele.

Nomenclatura dos pacotes

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- O **padrão** da **sun** para dar **nome** aos **pacotes** é relativo ao **nome** da **empresa** que **desenvolveu** a **classe**:
  - br.com.nomedaempresa.nomedoprojeto.subpacote
  - br.com.nomedaempresa.nomedoprojeto.subpacote2
  - br.com.nomedaempresa.nomedoprojeto.subpacote2.subpacote3
- Os **pacotes** só **possuem letras minúsculas**;
- As **classes** do **pacote padrão** de **bibliotecas** não seguem essa **nomenclatura**, que foi dada para **bibliotecas** de **terceiros**.

Import

- Para usar uma **classe** do **mesmo pacote**, basta fazer **referência** a ela como foi feito até **agora**, simplesmente **escrevendo** o **próprio nome** da **classe**.

- A **novidade** chega ao **tentar utilizar** a classe **Banco** (ou Cliente) em uma **outra classe** que esteja fora desse pacote, por exemplo, no pacote **br.com.caelum.util**:

```
package br.com.caelum.banco.util;

class TesteDoBanco {

    public static void main(String args[]) {
      br.com.caelum.banco.Banco meuBanco = new br.com.caelum.banco.Banco();
      meuBanco.nome = "Banco do Brasil";
      System.out.println(meuBanco.nome);
    }

}
```

Import

- Precisamos **referenciar** a classe **Banco** com **todo** o nome do **pacote** na sua **frente**. Esse é o conhecido ***Fully Qualified Name*** de uma classe

- Ao tentar **compilar** a classe **anterior**, surge um erro **reclamando** que a classe **Banco** não está **visível**.

- As classes são visíveis para outras no **mesmo pacote** e, para permitir que a classe **TesteDoBanco acesse** a classe **Banco** em outro **pacote**, precisamos **alterar** essa última e transformá-la em **pública**:

```
package br.com.caelum.banco;

public class Banco {
    String nome;
    Cliente clientes[] = new Cliente[2];
}
```

Import

- **Voltando** ao **código** do **TesteDoBanco**, é necessário **escrever** todo o **pacote** para identificar qual **classe queremos** usar? O exemplo que usamos ficou bem **complicado** de **ler**:

```
br.com.caelum.banco.Banco meuBanco = new br.com.caelum.banco.Banco();
```

- Existe uma **maneira** mais **simples** de se **referenciar** a classe **Banco**: basta **importá-la** do **pacote** br.com.caelum.banco:

```
package br.com.caelum.banco.util;

// para podermos referenciar
// a Banco diretamente
import br.com.caelum.banco.Banco;

class TesteDoBanco {

    public static void main(String args[]) {
        Banco meuBanco = new Banco();
        meuBanco.nome = "Banco do Brasil";
    }

}
```

Modificadores de Acesso

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Os **modificadores** de **acesso** existentes em **Java** são **quatro**, e até o **momento** já vimos **três**, mas só **explicamos dois**.

  - **public** – Todos podem **acessar aquilo** que for **definido** como **public**. **Classes**, **atributos**, **construtores** e **métodos** podem ser **public**.

  - **protected** – **Aquilo** que é **protected** pode ser **acessado** por **todas** as **classes** do mesmo **pacote** e por **todas** as **classes** que o **estendam**, mesmo que essas não estejam no mesmo pacote. Somente **atributos**, **construtores** e **métodos** podem ser **protected**.

Modificadores de Acesso

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- **padrão (sem nenhum modificador)** – Se **nenhum modificador** for **utilizado**, todas as **classes** do mesmo **pacote** têm **acesso** ao **atributo**, **construtor**, **método** ou **classe**.

- **private** – A **única classe** capaz de **acessar** os **atributos**, **construtores** e **métodos privados** é a **própria classe**. Classes, como conhecemos, **não podem** ser **private**, mas **atributos**, **construtores** e **métodos** sim.

Ferramenta – Jar

JAR

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

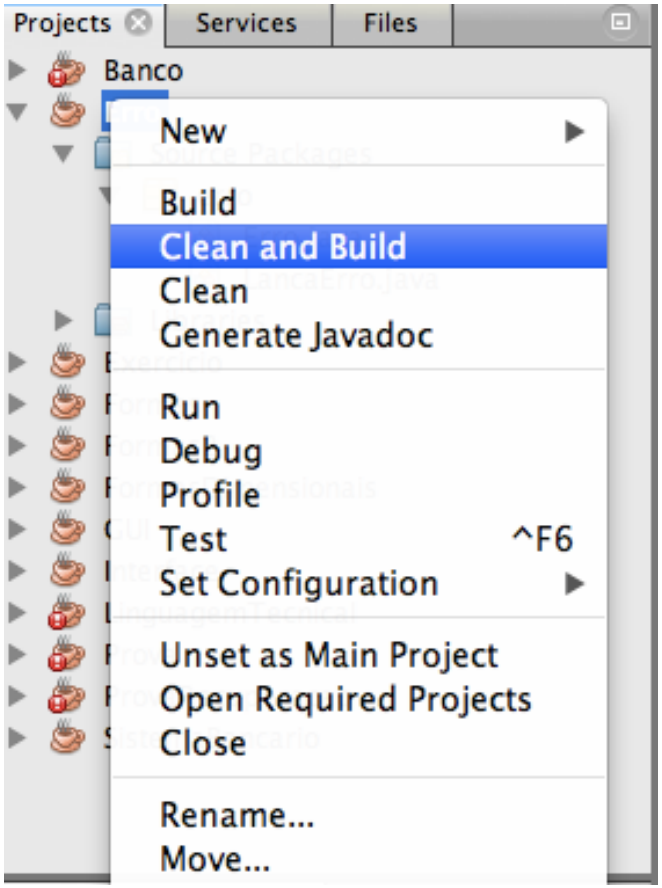
- Assim que um **programa** fica **pronto**, é meio **complicado** enviar **dezenas** ou **centenas** de **classes** para cada **cliente** que quer **utilizá-lo**.

- O **jeito** mais **simples** de **trabalhar** com um **conjunto** de **classes** é **compactá-los** em um **arquivo** só. O formato de **compactação padrão** é o **JAR**.

- O arquivo **jar** ou **Java ARchive,** possui um **conjunto** de **classes** (e arquivos de configurações) **compactados**, no **estilo** de um arquivo **zip**.

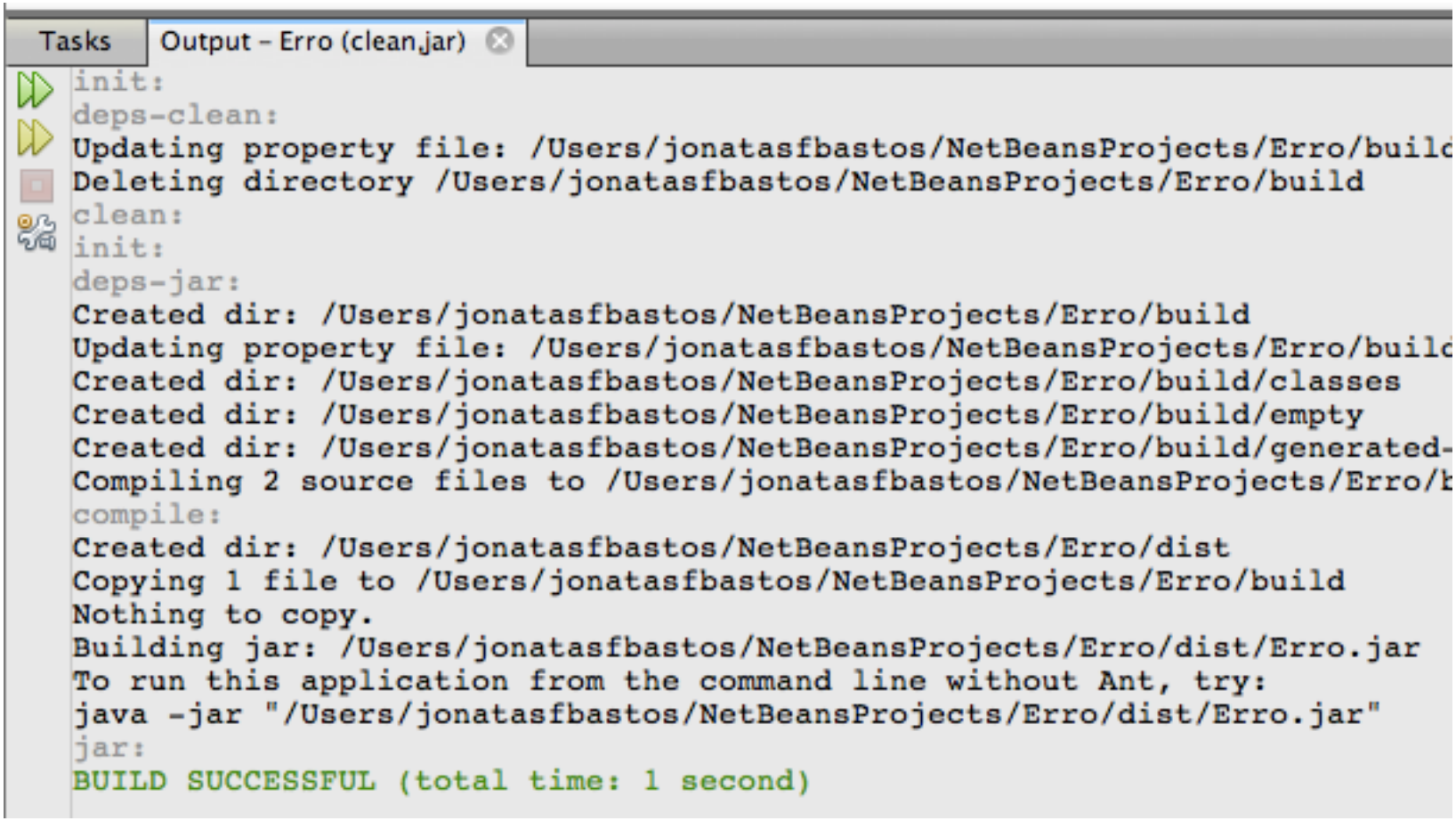
Gerando o JAR pelo NetBeans

- Primeiro **cllique** com o **botão direito** em cima do **nome** do seu **projeto** e **selecione** a **opção Clean and Build**.



Gerando o JAR pelo Eclipse

- Após **gerar** o **.jar** é só **explorar** a **pasta** onde ele foi **criado** e clicar em **cima** para **rodar** o **projeto**



Exercício