Suponha que temos os seguintes arquivos de propriedades e código. Quais valores são impressos nas linhas 8 e 9, respectivamente?

**Penguin.properties**

```
name=Billy
age=1
```

**Penguin_de.properties**

```
name=Chilly
age=4
```

**Penguin_en.properties**

```
name=Willy
```

```
5: Locale fr = new Locale("fr");
6: Locale.setDefault(new Locale("en", "US"));
7: var b = ResourceBundle.getBundle("Penguin", fr);
8: System.out.println(b.getString("name"));
9: System.out.println(b.getString("age"));
```

A. Billy e 1
B. Billy e null
C. Willy e 1
D. Willy e null
E. Chilly e null
F. O código não compila

---

**Análise da Questão**

**Locale**

O `Locale` em Java é uma classe que **representa uma região geográfica ou um idioma específico**. Ele é usado para internacionalização (i18n - abreviação para *internationalization* ), permitindo que aplicações Java se adaptem a diferentes idiomas e formatos culturais. Um `Locale` é composto por um código de idioma e, opcionalmente, um código de país.

**Criação de Locale** Você pode criar um `Locale` de várias maneiras:

1. **Somente idioma:**

   ```
   Locale fr = new Locale("fr");
   ```

   Aqui, estamos criando um `Locale` para o idioma francês. Este `Locale` não especifica um país.

2. **Idioma e país:**

```
Locale us = new Locale("en", "US");
```

Aqui, estamos criando um `Locale` para o idioma inglês nos Estados Unidos.

### ResourceBundle

O `ResourceBundle` é uma classe que **fornece um mecanismo para carregar diferentes conjuntos de recursos** (como mensagens e configurações) baseados na localidade (`Locale`). Esses recursos são geralmente armazenados em arquivos de propriedades (`.properties`).

### Métodos usados no código

1. **Locale.setDefault(Locale locale):**

   ```
   Locale.setDefault(new Locale("en", "US"));
   ```

   Este método define o `Locale` padrão da aplicação. No código do problema, estamos configurando o `Locale` padrão para inglês dos EUA.

2. **ResourceBundle.getBundle(String baseName, Locale locale):**

   ```
   var b = ResourceBundle.getBundle("Penguin", fr);
   ```

   Este método carrega um `ResourceBundle` baseado no nome base (neste caso, "Penguin") e no `Locale` especificado (neste caso, francês). Ele tenta encontrar o arquivo de propriedades que mais se adequa ao `Locale` **fornecido, seguindo uma hierarquia de fallback**.

### Hierarquia de Fallback do ResourceBundle

Quando você pede um `ResourceBundle` para um `Locale` específico, o Java segue uma hierarquia de fallback para encontrar o arquivo de propriedades mais apropriado:

1. Procura por um arquivo que **combine exatamente com o `Locale`**.
2. Se não encontrar, **tenta um arquivo que combine com o idioma e o país** (se especificado).
3. Se ainda não encontrar, **tenta um arquivo que combine somente com o idioma**.
4. Finalmente, se nenhum dos acima for encontrado, **usa o arquivo padrão (sem sufixos de idioma ou país**).

### Analisando o Código

Vamos rever o que acontece no código:

```
5: Locale fr = new Locale("fr");
6: Locale.setDefault(new Locale("en", "US"));
7: var b = ResourceBundle.getBundle("Penguin", fr);
```

```
8: System.out.println(b.getString("name"));
9: System.out.println(b.getString("age"));
```

1. **Linha 5:** Criamos um `Locale` para o francês (`fr`).
2. **Linha 6:** Configuramos o `Locale` padrão da aplicação para inglês dos EUA (`en_US`).
3. **Linha 7:** Carregamos o `ResourceBundle` chamado "Penguin" para o `Locale` francês (`fr`). O Java tenta encontrar o arquivo mais apropriado seguindo a hierarquia de fallback.
   - Procura por `Penguin_fr.properties` (não existe).
   - Procura por `Penguin_fr_FR.properties` (não existe).
   - Como o `Locale` padrão é `en_US`, procura por `Penguin_en.properties` e encontra.
4. **Linha 8:** Imprime o valor da chave `name` do `ResourceBundle`. Encontra `name=Willy` em `Penguin_en.properties`.
5. **Linha 9:** Imprime o valor da chave `age` do `ResourceBundle`. Como `Penguin_en.properties` não tem a chave `age`, ele busca no arquivo padrão `Penguin.properties` e encontra `age=1`.

Então, os valores impressos são `Willy` para o nome e `1` para a idade, resultando na resposta correta **C. Willy and 1**.