

- `void setSelected(boolean state)`

Pobiera lub ustawia stan elementu (`true` oznacza zaznaczony).

### 11.5.4. Menu podręczne

**Menu podręczne** (ang. *pop-up menu*) nie jest związane z paskiem menu, tylko pojawia się w różnych miejscach okna (rysunek 11.18).

**Rysunek 11.18.**  
Menu podręczne



Proces tworzenia menu podręcznego wygląda podobnie jak w przypadku zwykłego menu, z tym wyjątkiem, że nie nadaje mu się tytułu.

```
var popup = new JPopupMenu();
```

Elementy do takiego menu dodaje się w typowy sposób:

```
var item = new JMenuItem("Wytnij");
item.addActionListener(listener);
popup.add(item);
```

W przeciwieństwie do paska menu, który zawsze znajduje się na samej górze ramki, menu podręczne musi być wyświetlane za pomocą metody `show`. Należy w niej określić komponent nadrzędny menu oraz jego lokalizację za pomocą systemu współrzędnych komponentu nadrzędnego. Na przykład:

```
popup.show(panel, x, y);
```

Zazwyczaj menu podręczne są tak zaprogramowane, aby pokazywały się w odpowiedzi na kliknięcie przez użytkownika określonym przyciskiem myszy (tzw. *pop-up trigger*). W systemach Windows i Linux funkcję tę zazwyczaj pełni prawy przycisk myszy. Za pojawienie się menu kontekstowego w odpowiedzi na kliknięcie przez użytkownika przycisku wyzwalającego menu odpowiada poniższa instrukcja:

```
component.setComponentPopupMenu(popup);
```

Czasami do komponentu posiadającego menu kontekstowe może zostać wstawiony inny komponent, który również posiada takie menu. Komponent podrzędny może odziedziczyć menu kontekstowe elementu nadrzędnego dzięki poniższej instrukcji:

```
child.setInheritsPopupMenu(true);
```

`javax.swing.JPopupMenu` **1.2**

- `void show(Component c, int x, int y)`

Wyświetla menu kontekstowe nad komponentem `c`, umieszczając lewy górny róg tego menu w punkcie o współrzędnych  $(x, y)$ .

- `boolean isPopupTrigger(MouseEvent event)` **1.3**

Zwraca wartość `true`, jeśli zdarzenie myszy powoduje pojawienie się menu kontekstowego.

`java.awt.event.MouseEvent` **1.1**

- `boolean isPopupTrigger()`

Zwraca wartość `true`, jeśli zdarzenie myszy powoduje pojawienie się menu kontekstowego.

`javax.swing.JComponent` **1.2**

- `JPopupMenu getComponentPopupMenu()` **5.0**

- `void setComponentPopupMenu(JPopupMenu popup)` **5.0**

Pobiera lub ustawia menu kontekstowe dla komponentu.

- `boolean getInheritsPopupMenu()` **5.0**

- `void setInheritsPopupMenu(boolean b)` **5.0**

Pobiera lub ustawia własność `inheritsPopupMenu`. Jeśli własność ta jest ustawiona, a menu tego komponentu jest `null`, wykorzystuje menu kontekstowe jego komponentu nadrzędnego.

### 11.5.5. Mnemoniki i akceleratory

Dla zaawansowanego użytkownika programu bardzo ważnym usprawnieniem pracy jest możliwość otwierania menu za pomocą **mnemoników**. Mnemoniki do elementów menu określa się poprzez określenie wybranej litery w konstruktorach tych elementów:

```
var aboutItem = new JMenuItem("O programie", 'O');
```

Mnemonic jest wyświetlany automatycznie w menu, a litera mnemonicu jest podkreślona (rysunek 11.19). Na przykład etykieta elementu zdefiniowanego powyżej będzie wyglądała następująco: *O programie* — podkreślona litera *O*. Po rozwinięciu menu wystarczy nacisnąć klawisz *O*, aby wybrać ten element (jeśli litera mnemonicu nie występuje w łańcuchu menu, jej naciśnięcie spowoduje wybór tego elementu, ale mnemonic nie będzie wyświetlany w menu — oczywiście przydatność takich niewidocznych mnemoniców stoi pod znakiem zapytania).

**Rysunek 11.19.**  
Mnemoniki



Czasami programista nie chce, aby podkreślona została pierwsza litera pasująca do mnemonicu. Jeśli mamy na przykład mnemonic *A* dla elementu menu *Zapisz jako*, możemy sprawić, aby została podkreślona litera *a* w drugim wyrazie (*Zapisz jako*). W Java SE 1.4 wprowadzono