

Przykład 4.25. Poszukiwanie tekstu

```
type
  TSearchFuture = class(TFuture)
  private
    fEditorText: string;
    fFindPos: LongInt;
    fFindText: string;
    fOptions: TFindOptions;
    procedure FindDown(out Result: Variant);
    procedure FindUp(out Result: Variant);
  public
    constructor Create(Editor: TRichEdit; Options: TFindOptions;
      const Text: string);
    function Compute: Variant; override;

    property EditorText: string read fEditorText;
    property FindPos: LongInt read fFindPos write fFindPos;
    property FindText: string read fFindText;
    property Options: TFindOptions read fOptions;
  end;

{ TSearchFuture }

constructor TSearchFuture.Create(Editor: TRichEdit;
  Options: TFindOptions; const Text: string);
begin
  inherited Create;
  TimeOut := 30000; // Limit 30 sekund na znalezienie szukanego
  // tekstu.

  // Zapamiętanie podstawowych parametrów wyszukiwania.
  fEditorText := Editor.Text;
  fOptions := Options;
  fFindText := Text;
  // Rozpoczęcie szukania na końcu bieżącego podświetlenia,
  // aby uniknąć odnajdywania ciągle tego samego tekstu.
  fFindPos := Editor.SelStart + Editor.SelLength;
end;
//-----
// Wyszukiwanie tekstu.
function TSearchFuture.Compute: Variant;
begin
  if not (frMatchCase in Options) then
  begin
    fFindText := AnsiLowerCase(FindText);
    fEditorText := AnsiLowerCase(EditorText);
  end;

  if frDown in Options then
    FindDown(Result)
  else
    FindUp(Result);
end;
//-----
procedure TSearchFuture.FindDown(out Result: Variant);
var
  Next: PChar;
begin
  // Znajdź następny pasujący fragment.
  Next := AnsiStrPos(PChar(EditorText) + FindPos, PChar(FindText));
  if Next = nil then
    // Nie znaleziono.
    Result := -1
```

```

else
begin
    // Znaleziono: zwrócenie pozycji startowej odnalezionego tekstu.
    FindPos := Next - PChar(EditorText);
    Result := FindPos;
end;
end;
//-----
procedure TSearchFuture.FindUp(out Result: Variant);
var
    Previous: PChar;
begin
    // Znajdź poprzedni pasujący fragment.
    Previous := AnsiStrPos(PChar(EditorText) - FindPos,
        PChar(FindText));
    if Previous = nil then
        // Nie znaleziono.
        Result := -1
    else
        begin
            // Znaleziono: zwrócenie pozycji startowej odnalezionego tekstu.
            FindPos := Previous - PChar(EditorText);
            Result := FindPos;
        end;
end;
end;

```

Jeżeli użytkownik przechodzi do edycji pliku, zmienia podświetlenie lub w inny sposób czyni nieaktualnym bieżące poszukiwanie, edytor wstrzymuje działanie obiektu transakcji, ponieważ staje się ono logicznie niepoprawne. Kiedy użytkownik rozpocznie nowy proces poszukiwania, edytor musi wystartować zupełnie nowy obiekt transakcji. Przykład 4.26 przedstawia odpowiednie metody dla formularza TMDIChild.

Przykład 4.26. Zarządzenie obiektem Future edytora tekstu

```

var fFuture: TFuture;
    Editor: TRichEdit;
...
//-----
// Restart procesu poszukiwania w tle, jeżeli użytkownik zmieni
// podświetlenie.
procedure TMDIChild.EditorSelectionChange(Sender: TObject);
begin
    if Future <> nil then
        RestartSearch;
end;
//-----
// Zatrzymanie wątku pracującego w tle, jeśli użytkownik zamknie
// okno dialogowe "Znajdź"
procedure TMDIChild.FindDialogClose(Sender: TObject);
begin
    FreeAndNil(fFuture);
end;
// Restart procesu poszukiwania w przypadku zmiany parametrów
// poszukiwania.
//-----
procedure TMDIChild.RestartSearch;
begin
    FreeAndNil(fFuture);
    fFuture := TSearchFuture.Create(Editor, FindDialog.Options,
        FindDialog.FindText);
end;

```