

Bezpieczne uruchamianie apletów

wg

<http://java.sun.com/docs/books/tutorial/security1.2/>

Zabezpieczenia przed uruchamianiem apletów na pisanych przez nieznanych autorów

- 1) ograniczenie możliwości odczytywania, zapisywania i usuwania plików z dysku komputera użytkownika apletu
- 2) pobieranie informacji o plikach dyskowych komputera użytkownika apletu

Pierwszy sposób zabezpieczania apletów

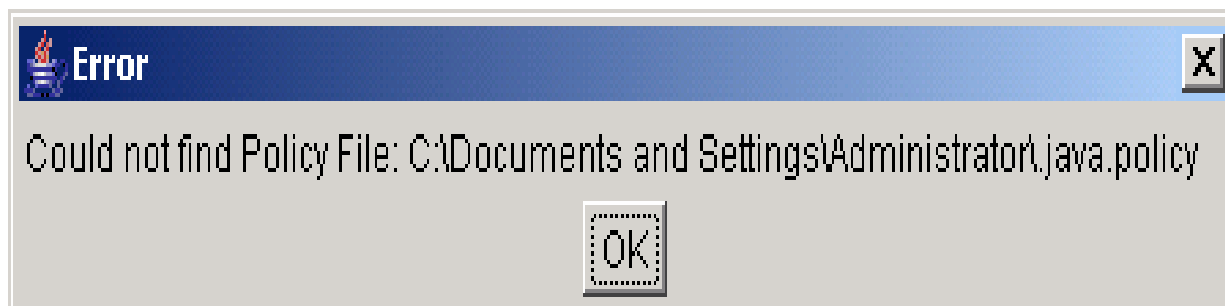
Atak przed wirusami ze strony apletów zabezpiecza **security manager**, który zainstalowany domyślnie nie pozwala na operacje plikowe w aplecie-
konfigurowanie za pomocą narzędzia **PolicyTool**.

Drugi sposób zabezpieczenia apletów - podpisy cyfrowe:

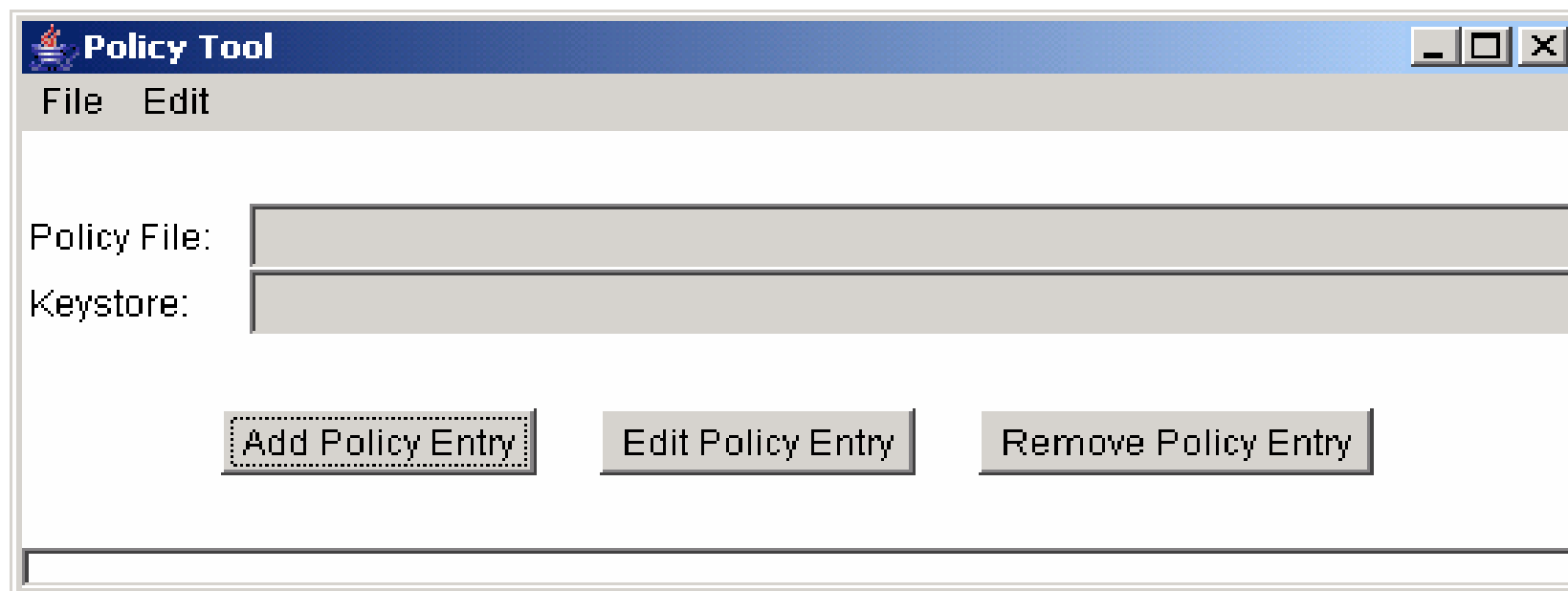
- a) generowanie klucza publicznego i prywatnego przez autora apletu za pomocą narzędzia **keytool**
- b) otrzymanie przez autora apletu **certyfikatu** od firmy wydającej certyfikaty bezpieczeństwa na podstawie klucza głównego
- c) złożenie **podpisu cyfrowego** w aplecie (w pliku typu jar) przez autora apletu wygenerowanego na podstawie certyfikatu, klucza głównego i prywatnego
- d) użytkownik na podstawie podpisu cyfrowego dołączonego do apletu może poszerzyć uprawnienia apletu za pomocą narzędzia **jarsigner**

Pierwszy sposób jest realizowany za pomocą PolicyTool

a) Uruchomienie narzędzia **PolicyTool** - komunikat świadczy o braku pliku domyślnego we wskazanym katalogu



b) Należy nacisnąć przycisk **Add Policy Entry**. Specyfikuje ona jeden lub więcej przywilejów dla apletów z wyszczególnionego URL.



c) Pola **CodeBase** i **SignedBy** są używane do specyfikowania kodu godnego zaufania do wykonywania operacji np. plikowych.

CodeBase: wskazanie położenia kodu apletu (adres URL); pole puste, oznacza brak ograniczeń

SignedBy: oznacza alias certyfikatu powiązanego z kluczem publicznym, które pozwalają zweryfikować wiarygodność apletu zawierającego podpis cyfrowy utworzony na podstawie klucza głównego i prywatnego oraz tego certyfikatu przez twórcę apletu. Wejście *SignedBy* jest opcjonalne; pominięcie go oznacza brak potrzeby uwierzytelnienia kodu

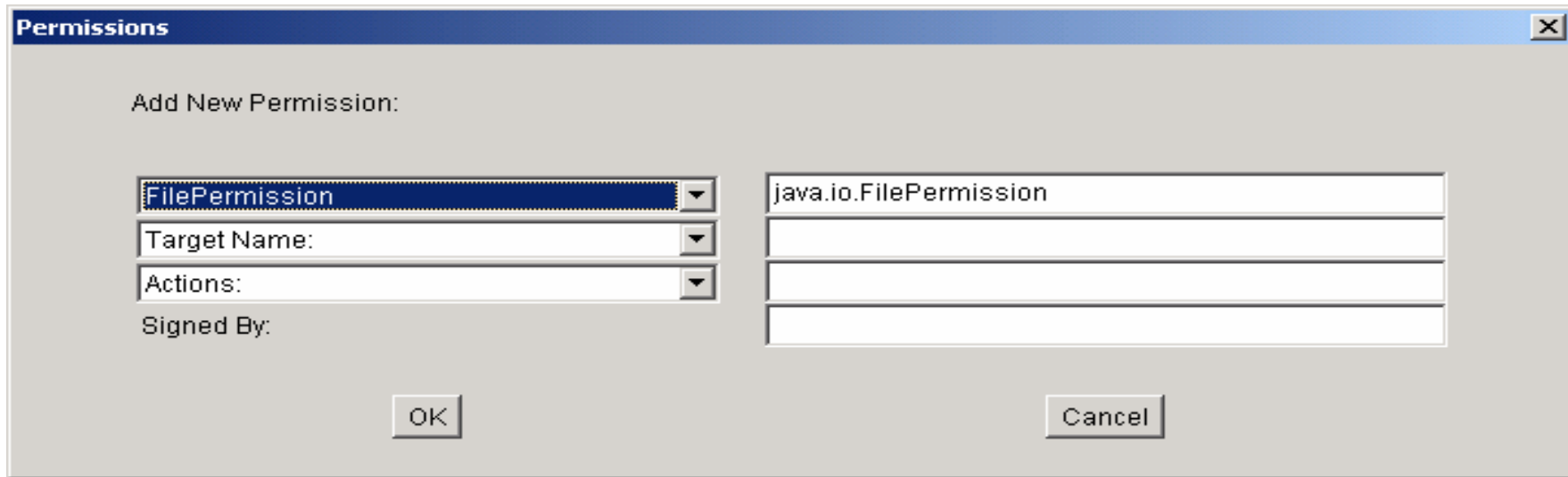
The image shows a 'Policy Entry' dialog box. The 'CodeBase' field contains the URL 'http://sprocket.ict.pwr.wroc.pl/~zkruczkiewicz/wyklady/java'. The 'SignedBy' field is empty. There are three buttons for managing principals: 'Add Principal', 'Edit Principal', and 'Remove Principal'. Below these are three buttons for managing permissions: 'Add Permission', 'Edit Permission', and 'Remove Permission'. At the bottom are 'Done' and 'Cancel' buttons.

d) Po wypełnieniu pola CodeBase adresem URL i braku SignedBy należy przycisnąć przycisk **Add Permission** i wypełnić trzy pola pokazanego formularza

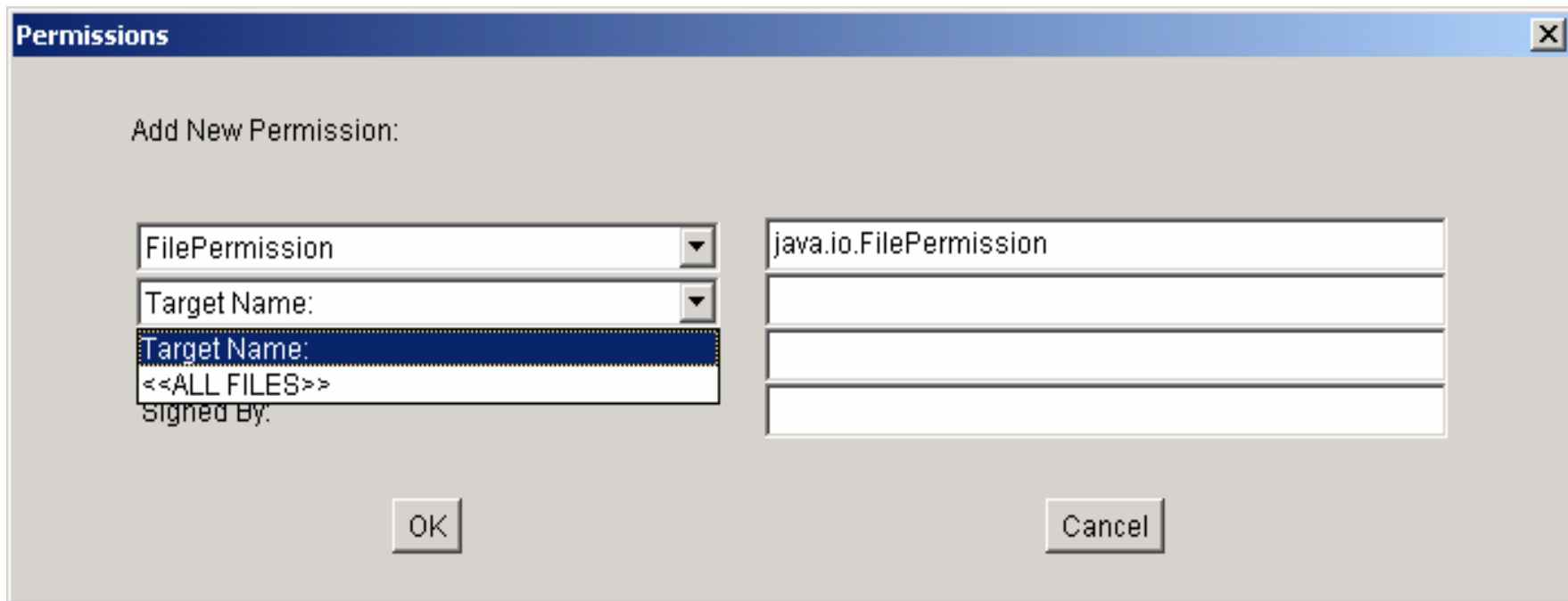
The screenshot shows a dialog box titled "Permissions" with a close button (X) in the top right corner. Below the title bar, the text "Add New Permission:" is displayed. There are four input fields on the left side, each with a dropdown arrow: "Permission:", "Target Name:", "Actions:", and "Signed By:". To the right of these fields are four empty text input boxes. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

This screenshot shows the same "Permissions" dialog box, but with the "Permission:" dropdown menu open. The list of permissions includes: "Permission:", "AllPermission", "AudioPermission", "AuthPermission", "AWTPermission", "DelegationPermission", "FilePermission" (which is highlighted in blue), and "LoggingPermission". The "Signed By:" field and the "Cancel" button are also visible.

e) Wybrano pozwolenie na operacje plikowe



f) Wybór nazwy pliku do zapisu lub odczytu przez aplet



g) Podano plik o nazwie *Testplik.dat*

Permissions

Add New Permission:

FilePermission
Target Name: Testplik.dat
Actions:
Signed By:

java.io.FilePermission
Testplik.dat

OK Cancel

h) Wybór operacji np. zapisu do pliku *Testplik.dat*

Permissions

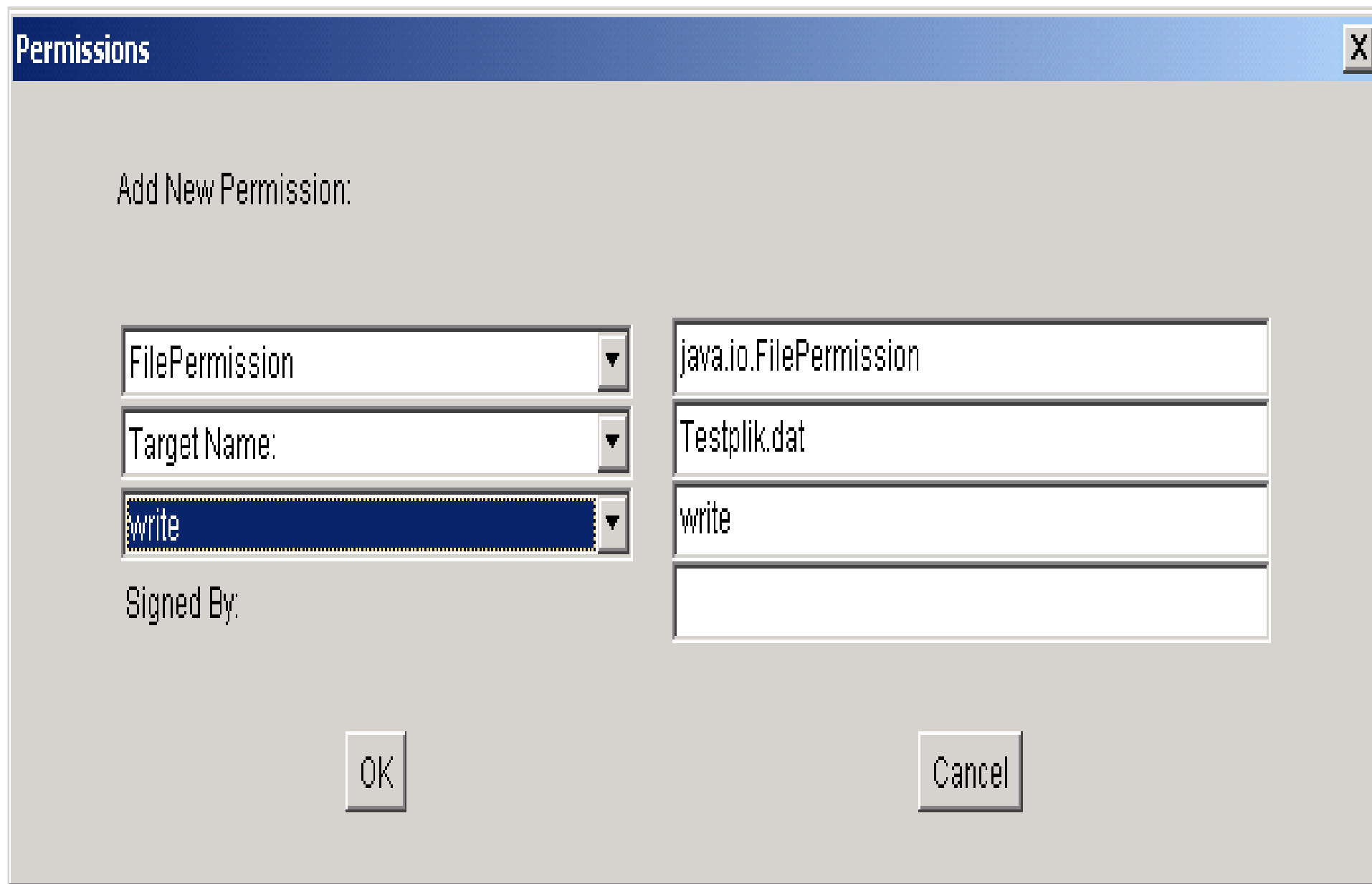
Add New Permission:

FilePermission
Target Name: Testplik.dat
Actions:
Actions:
read
write
delete
execute
read, write, delete, execute

java.io.FilePermission
Testplik.dat

Cancel

i) Pełna specyfikacja operacji wykonywanych przez aplet na komputerze przeglądarki. Należy nacisnąć klawisz **OK**



The image shows a 'Permissions' dialog box with a blue title bar and a close button (X) in the top right corner. The main area is light gray. It contains the following elements:

- Add New Permission:** A label above a set of three dropdown menus.
- FilePermission:** A dropdown menu with 'FilePermission' selected.
- Target Name:** A dropdown menu with 'Testplik.dat' selected.
- write:** A dropdown menu with 'write' selected and highlighted in blue.
- Signed By:** A label above an empty text input field.
- java.io.FilePermission:** A text input field containing 'java.io.FilePermission'.
- Testplik.dat:** A text input field containing 'Testplik.dat'.
- write:** A text input field containing 'write'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

j) Po przejściu do kolejnego okna zatwierdzamy opcje bezpieczeństwa za pomocą klawisza **Done**

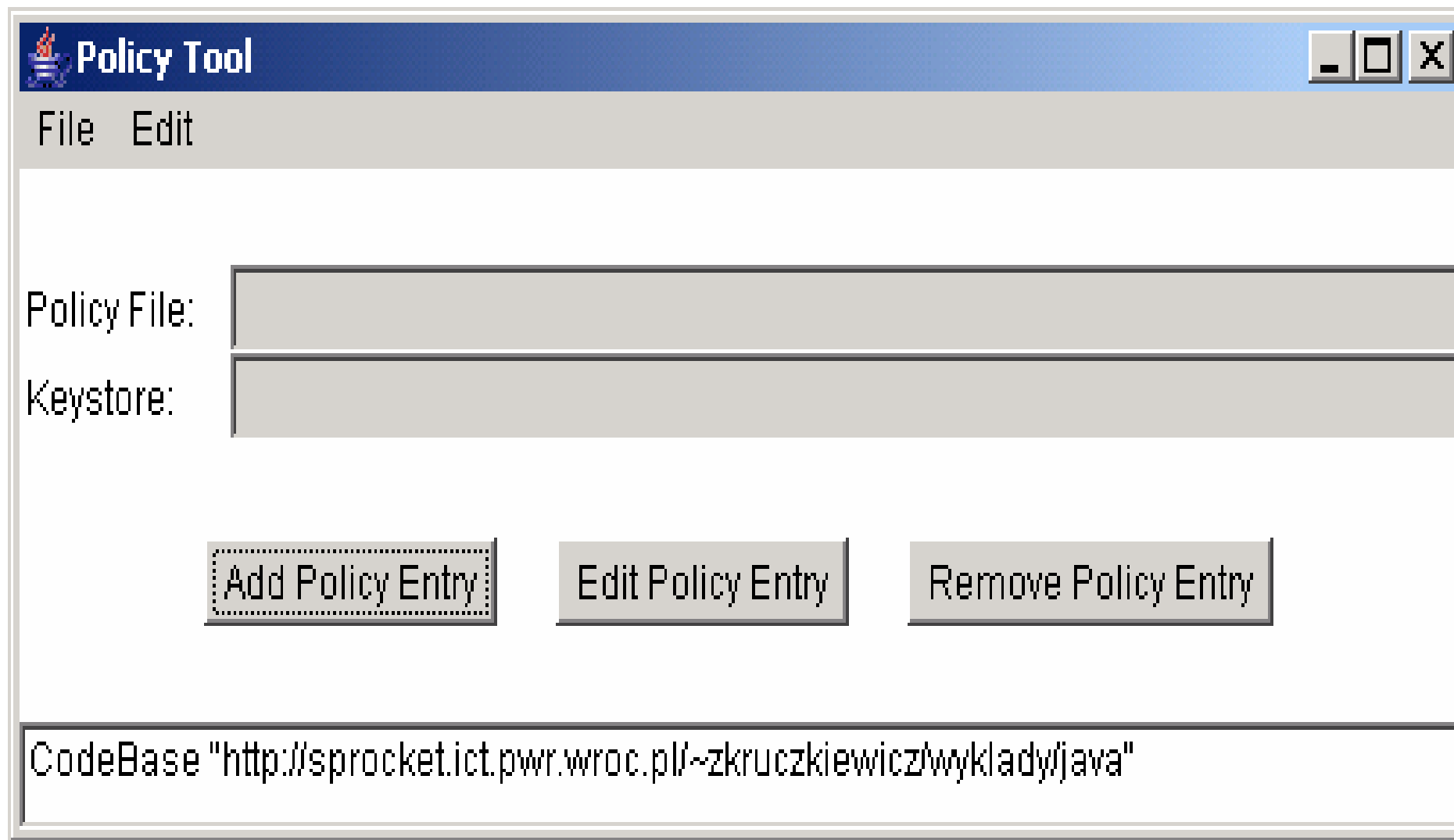
Policy Entry

CodeBase:

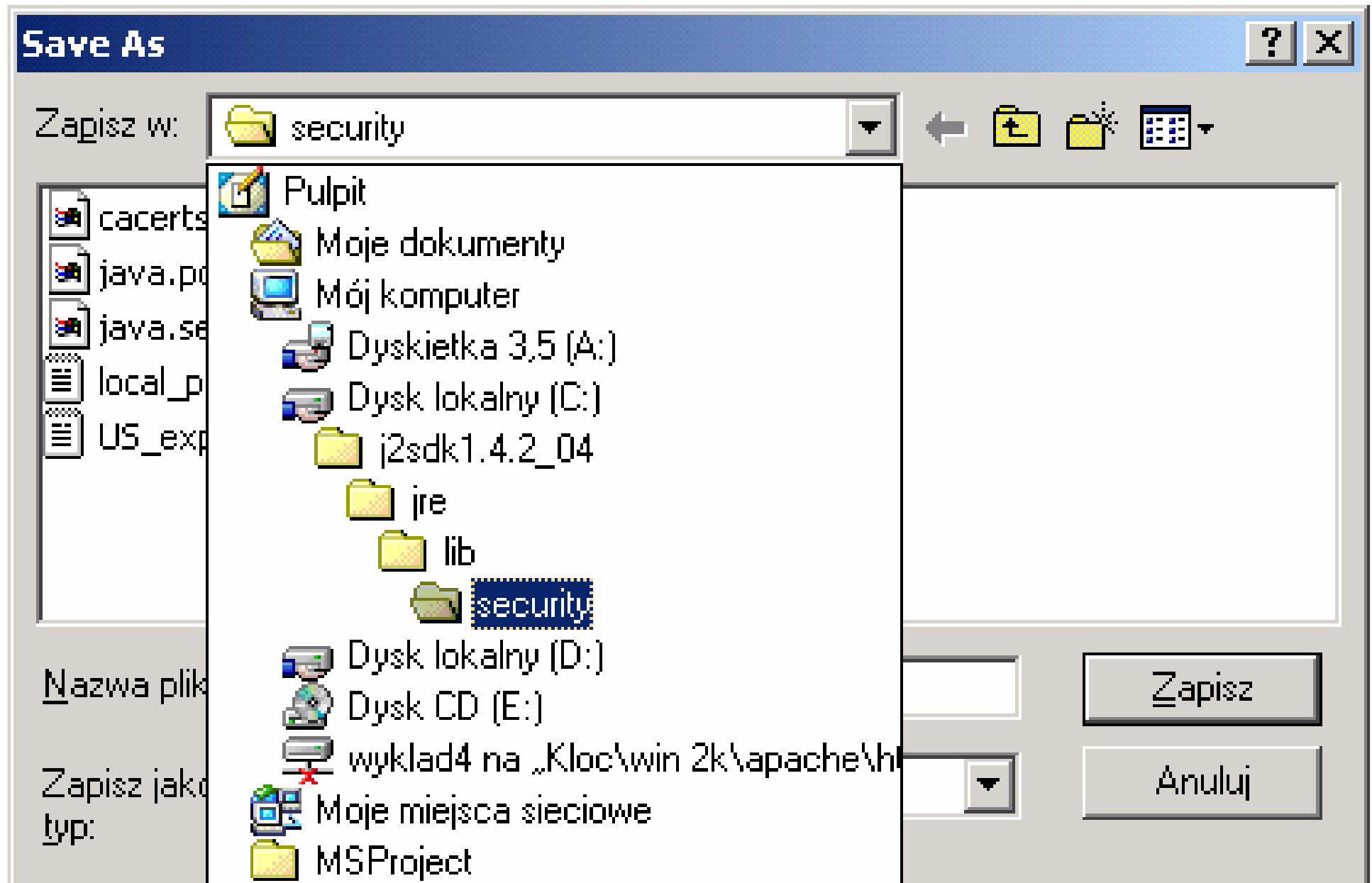
SignedBy:

Principals:

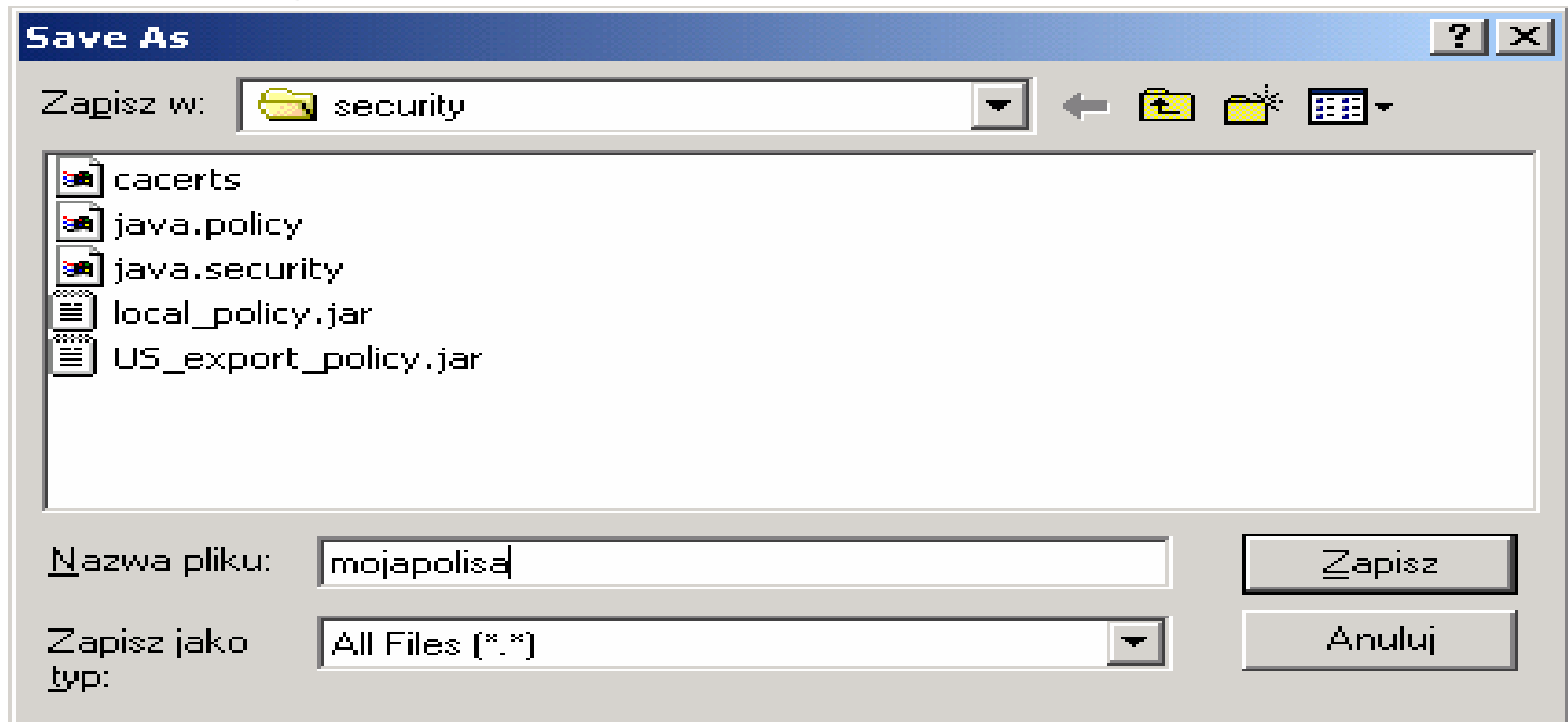
k) Należy teraz wybrać z opcji **File** opcję **Save As**



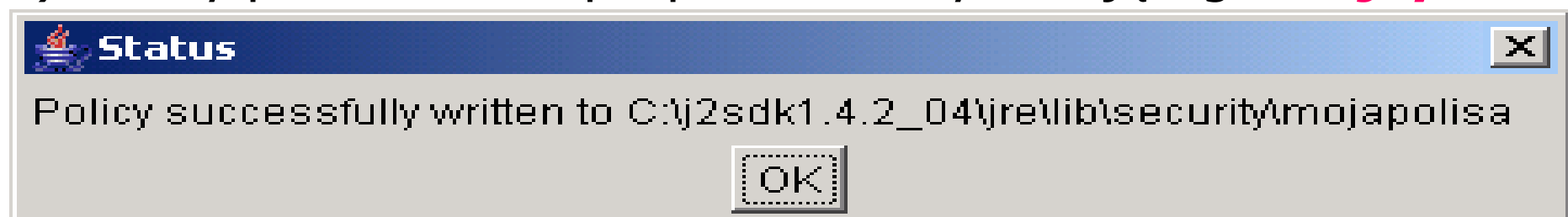
l) Po wyborze **Save As** należy wybrać katalog do zapisu pliku uwierzytelniającego



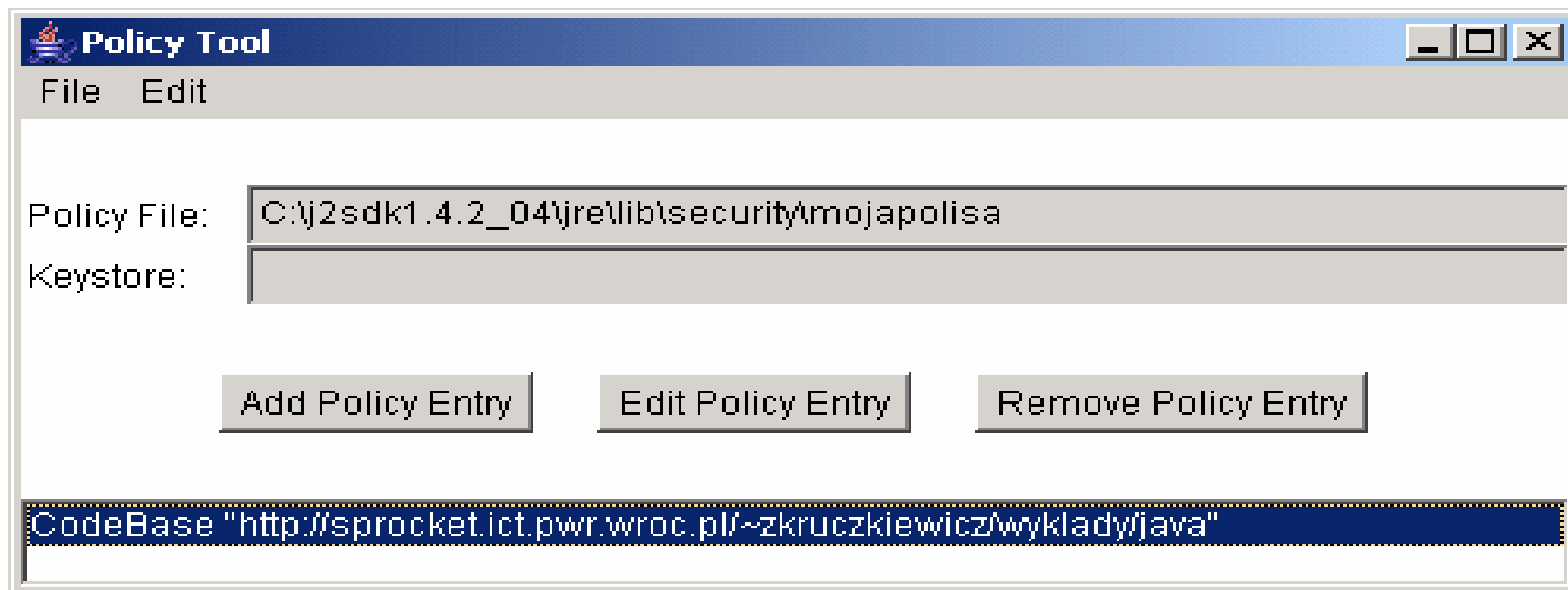
m) Należy zapisać plik uwierzytelniający po wypełnieniu pola tekstowego **Nazwa pliku** i zatwierdzeniu przyciskiem **Zapisz**



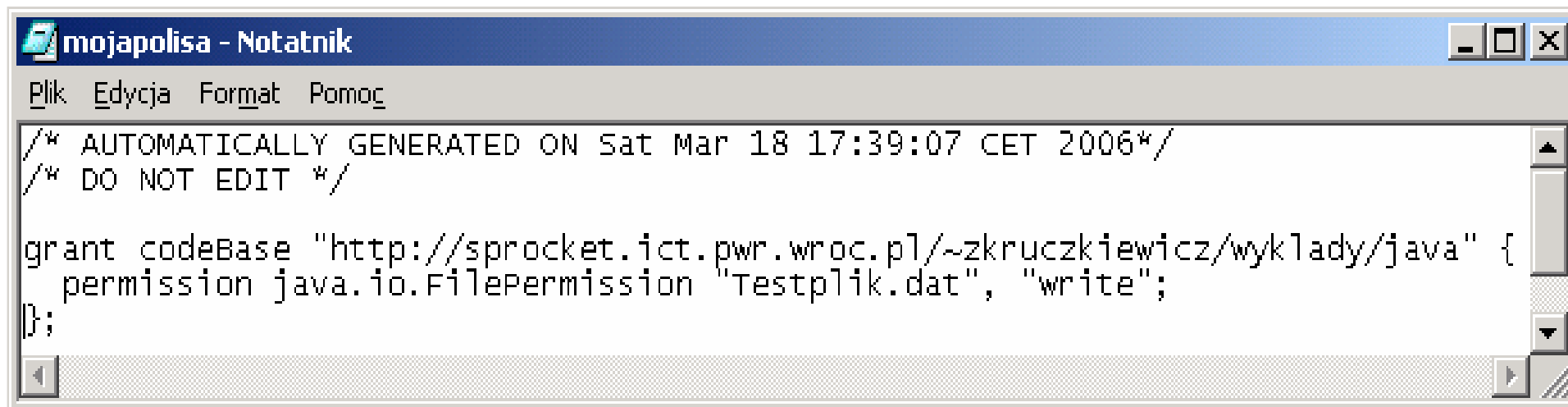
n) Należy potwierdzić zapis pliku uwierzytelniającego ***mojapolisa***



o) Narzędzie **PolicyTool** należy zamknąć za pomocą opcji **Exit** z menu **File**



p) Zawartość utworzonego pliku uwierzytelniającego *mojapolisa*



Dwa sposoby wykorzystania pliku uwierzytelniającego *mojapolisa*

1) Pierwszy sposób: uruchamianie apletu w katalogu bieżącym, w którym umieszczono plik *mojapolisa* (jedna linia poleceń)

```
C:\j2sdk1.4.2_04\bin\appletviewer  
-J-Djava.security.policy=mojapolisa  
http://sprocket.ict.pwr.wroc.pl/~zkruczkiewicz/  
wyklady/java/Testpliki.html
```

2) Drugi sposób: uruchomienie apletu o danym adresie URL niezależnie od położenia pliku uwierzytelniającego *mojapolisa*
Uwaga: jeśli z tej wersji Java korzysta przeglądarka, ten sposób jest również wykorzystany podczas uruchamiania apletu przez przeglądarkę.

C:\j2sdk1.4.2_04\bin\appletviewer

**http://sprocket.ict.pwr.wroc.pl/~zkruczkiewicz/
wyklady/java/Testpliki.html**

Po wcześniejszej modyfikacji zawartości pliku java.security :

Windows: java.home\lib\security\java.security

UNIX: java.home/lib/security/java.security

**# The default is to have a single system-wide policy file,
and a policy file in the user's home directory.**

policy.url.1=file:\${java.home}/lib/security/java.policy

policy.url.2=file:\${user.home}/.java.policy

#tutaj dopisano dostęp do nowego pliku typu policy

policy.url.3=file:\${java.home}/lib/security/mojapolisa

Przykład apletu realizującego czynności plikowe

// próba zapisu i odczytu przez aplet *Testpliki* pliku o innej nazwie niż
// podano w pliku uwierzytelniającym, czyli *Testplik* zamiast
// *Testplik.dat*

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Testpliki extends JApplet
```

```
{
```

```
    static byte weByte()
```

```
    { InputStreamReader wejscie = new InputStreamReader( System.in );
```

```
      BufferedReader bufor = new BufferedReader( wejscie );
```

```
      StringTokenizer zeton;
```

```
      try
```

```
      { zeton = new StringTokenizer(bufor.readLine());
```

```
        return Byte.parseByte(zeton.nextToken()); } 
```

```
      catch (Exception e)
```

```
      { System.err.println("Bład typu "+e);
```

```
        return 0; } 
```

```
}
```



```

static void Zapiszplik2()
{
    int dane=0;
    try
    {FileOutputStream plik = new FileOutputStream ("Testplik");
      BufferedOutputStream bufor = new BufferedOutputStream (plik);
      while (dane!=-1)
      { System.out.print("Podaj dane: ");
        dane=weByte();
        if (dane!=-1)
            bufor.write(dane);
        }
      bufor.close();
    } catch (IOException e)
      { System.out.println ("Bład zapisu pliku bajtowego"+e); }
}

```

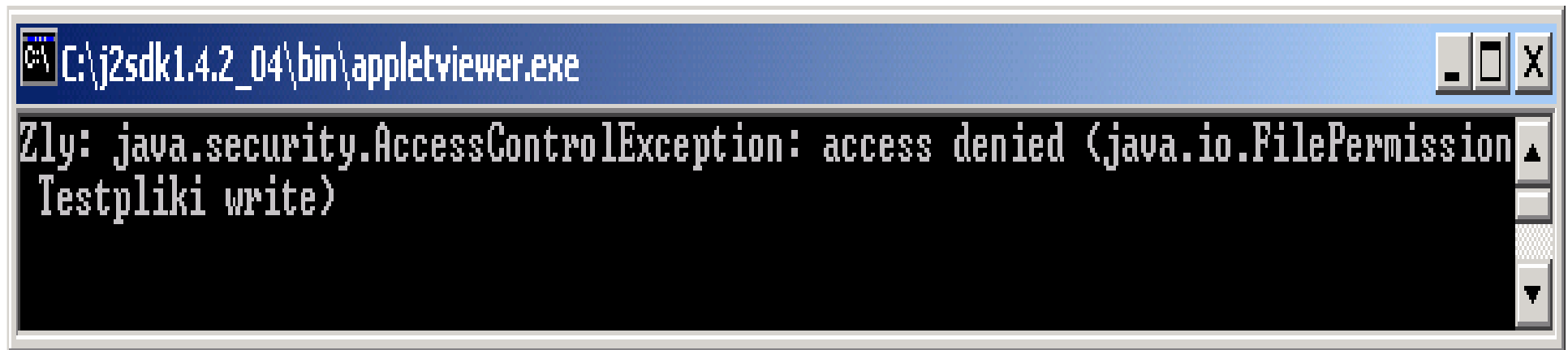
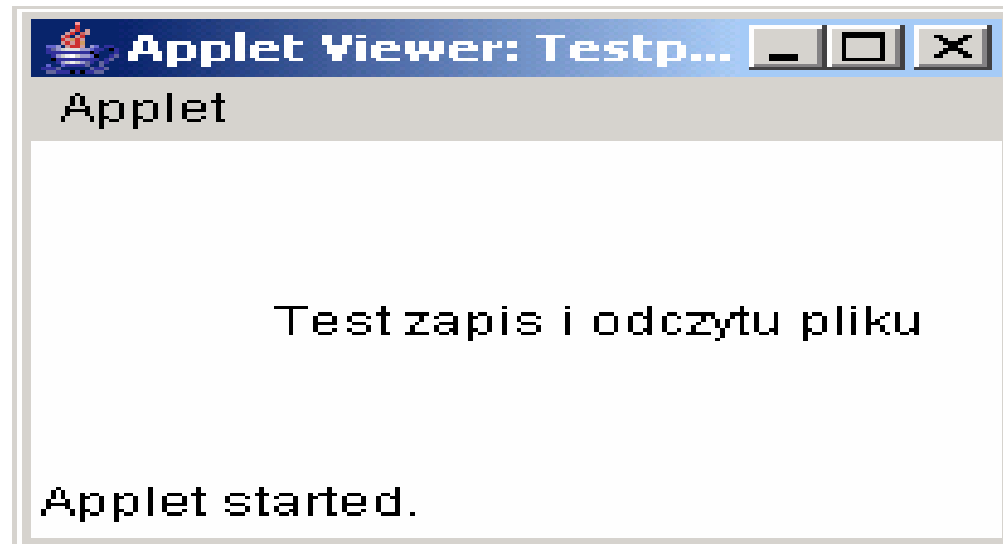
```

static void Odczytajplik2()
{ int dane=0;
  try
  { FileInputStream plik = new FileInputStream ("Testplik");
    BufferedInputStream bufor = new BufferedInputStream (plik);
    dane=plik.read();
    while (dane!=-1)
    { System.out.print(dane);
      dane=bufor.read(); }
    System.out.println("");
    bufor.close();
  } catch (IOException e)
    { System.out.println ("Bład odczytu pliku bajtowego"+e); }
}

public void paint(Graphics g)
{ g.drawString("Test zapis i odczytu pliku", 50, 60 );
  try
  { Zapiszplik2();
    Odczytajplik2();
  } catch (Exception e)
    { System.out.println("Zły: "+e); }
}}

```

Skutki braku uwierzytelnienia apletu po uruchomieniu przez appletviewer – nazwa pliku zapisywanego przez aplet jest niezgodna z nazwą podaną w pliku *mojapolisa*



Po właściwym przygotowaniu pliku uwierzytelniającego
mojapolisa

Policy Entry

CodeBase:

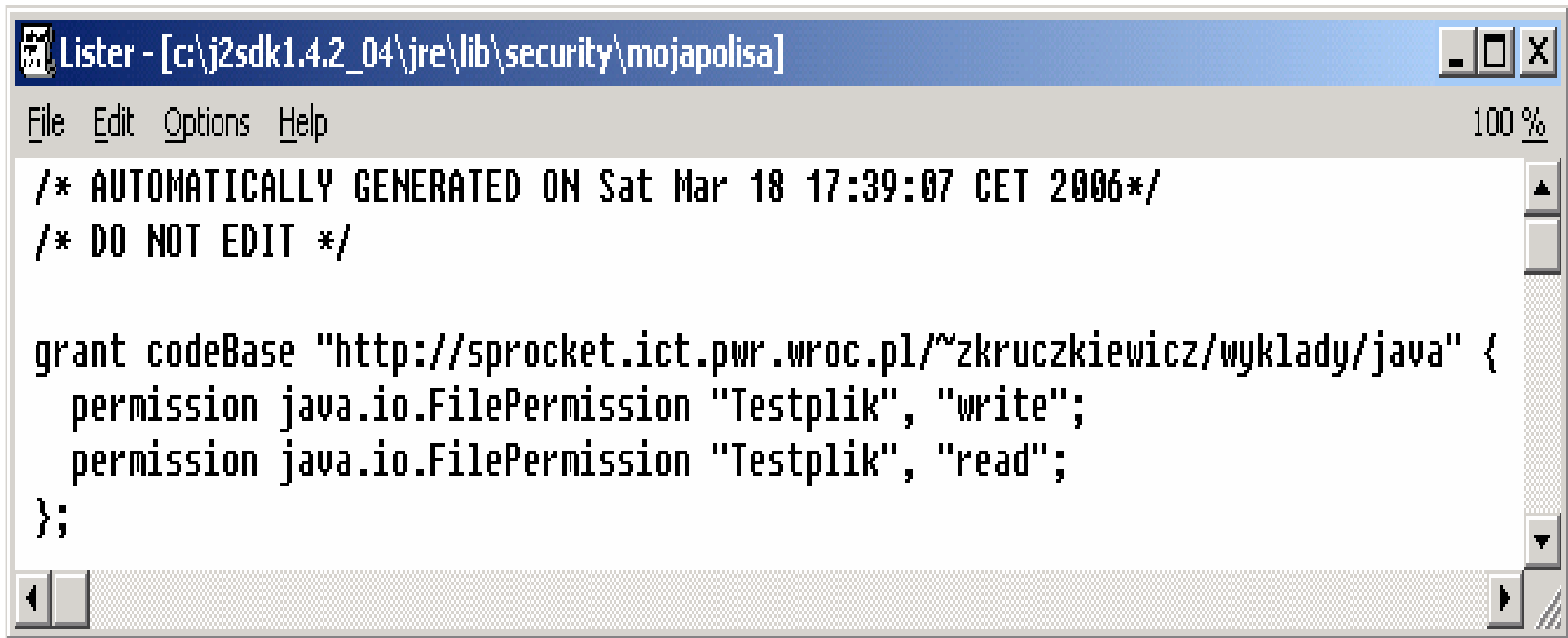
SignedBy:

Principals:

```
permission java.io.FilePermission "Testplik", "write";  
permission java.io.FilePermission "Testplik", "read";
```

Nowa zawartość pliku *mojapolisa*, który pozwala apletom umieszczonym pod adresem URL:

<http://sprocket.ict.pwr.wroc.pl/~zkruczkiewicz/wyklady/java> zapisywać i odczytywać po stronie klienta w katalogu bieżącym plik o nazwie *Testplik*.



The screenshot shows a Notepad window titled "Lister - [c:\j2sdk1.4.2_04\jre\lib\security\mojapolisa]". The window contains the following text:

```
File Edit Options Help 100 %
/* AUTOMATICALLY GENERATED ON Sat Mar 18 17:39:07 CET 2006*/
/* DO NOT EDIT */

grant codeBase "http://sprocket.ict.pwr.wroc.pl/~zkruczkiewicz/wyklady/java" {
    permission java.io.FilePermission "Testplik", "write";
    permission java.io.FilePermission "Testplik", "read";
};
```

Tak działa aplet po skonfigurowaniu menedżera bezpieczeństwa, który pozwala mu zapisywać i odczytywać w katalogu bieżącym plik o nazwie *Testplik*



```
C:\j2sdk1.4.2_04\bin\appletviewer.exe
Podaj dane: 1
Podaj dane: 2
Podaj dane: 3
Podaj dane: 4
Podaj dane: 5
Podaj dane: 6
Podaj dane: 7
Podaj dane: -1
1234567
Podaj dane: _
```

