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

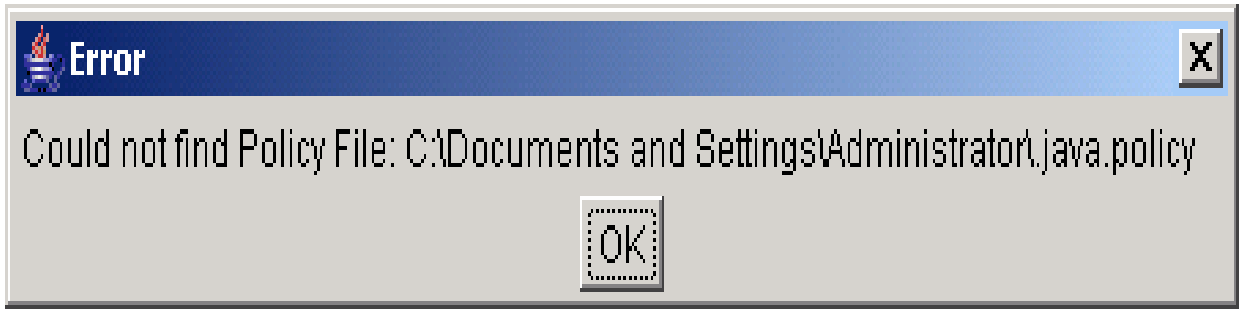
Atak przed wirusami ze strony apletów zabezpiecza **security manager,** który zainstalowany domyślnie nie pozwala na operacie plikowe w aplecie- konfigurowanie za pomocą narzędzia **PolicyTool**.

Drugi sposób zabezpieczenia apletów - podpisy cyfrowe:

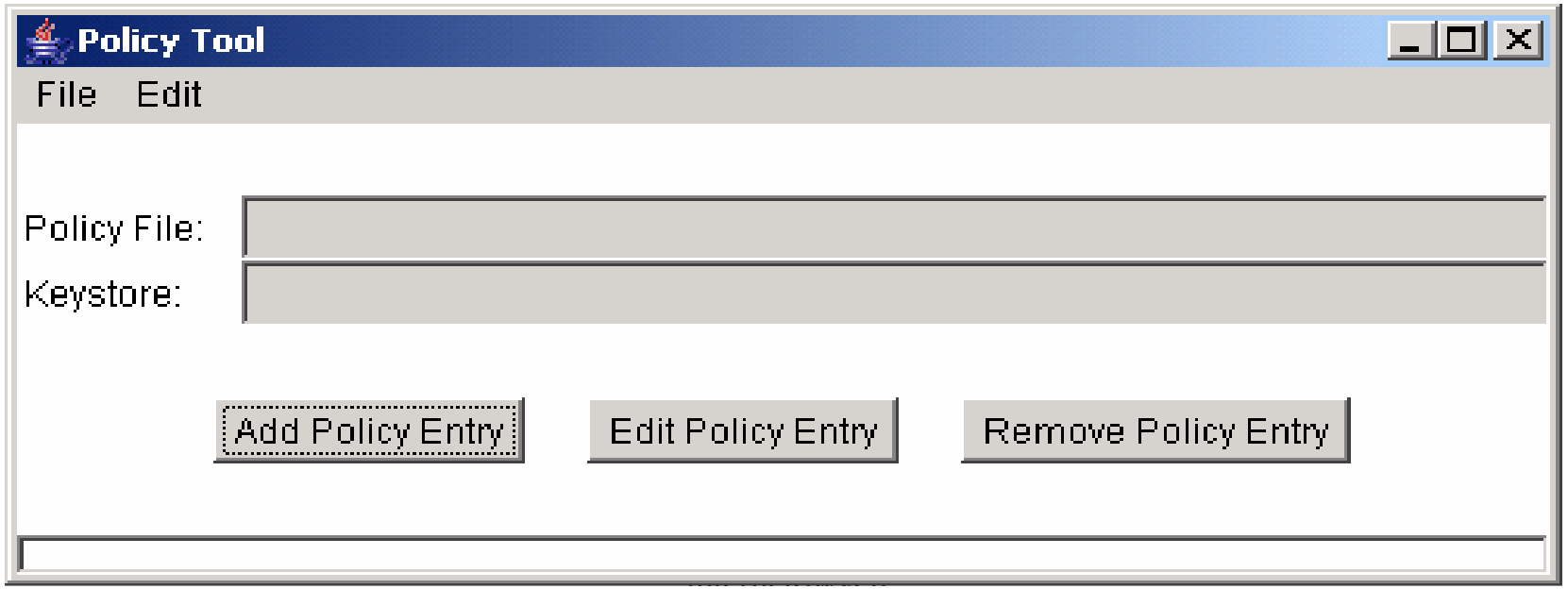
a) generowanie klucza publicznego i prywatnego przez autora apletu za pomocą narzędzia **keytool**
b) otrzymanie przez autora apletu **certyfikatu** od firmy wydającej certyfikaty bezpieczeństwa na podstawie klucza głównego
c) złożenie **podpisu cyfrowego** w aplecie (w pliku typu jar) przez autora apletu 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



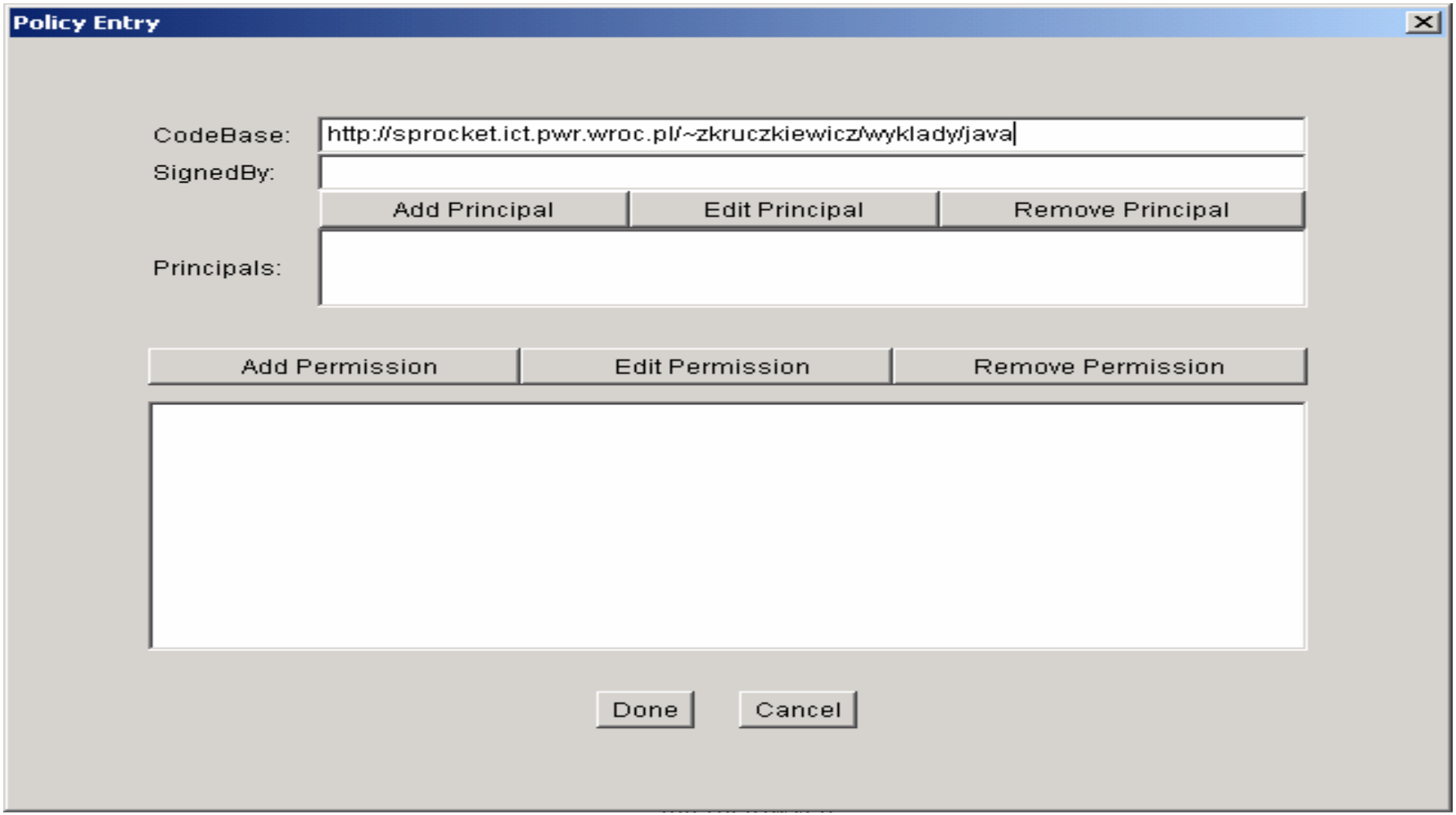
b) Należy nacisnąć przycisk **Add Policy Entry**. Specyfikuje ona jeden lub więcej przywilejów dla apletów z wyszczególnionego URL.



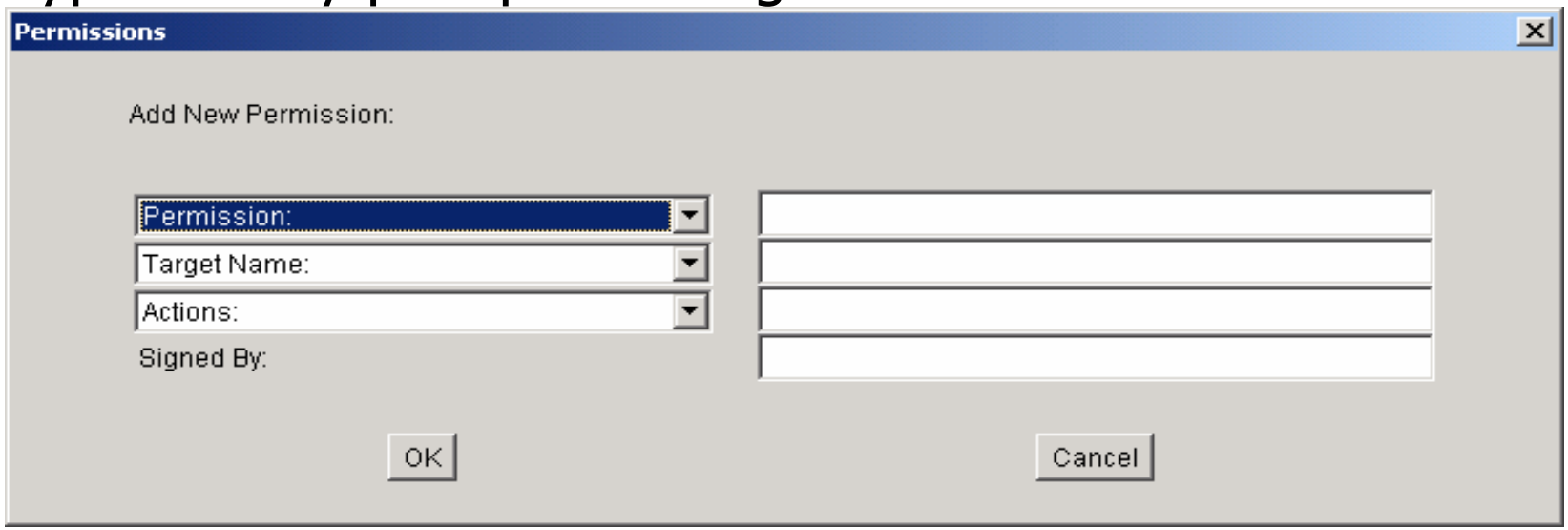
c) Pola **CodeBase** i **SignedBy** są używane do specyfikowania kodu godnego zaufania do wykonywania operacji np. plikowych.

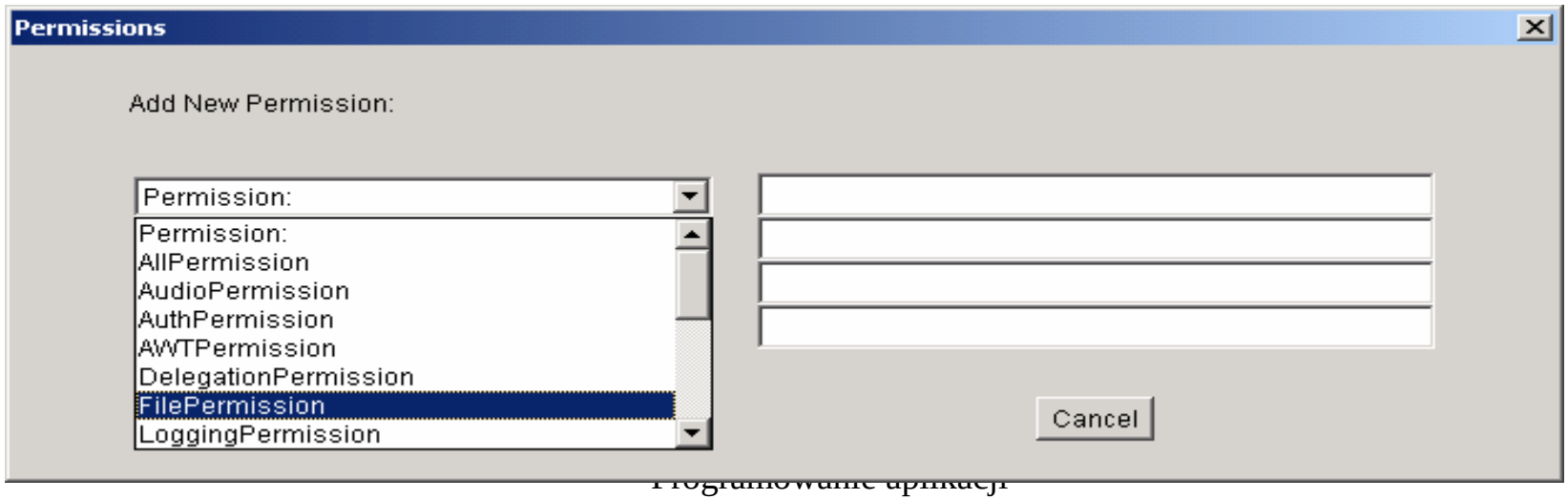
CodeBase: wskazanie położenia kodu apletu (adres URL); pole puste, oznacza brak ograniczeń

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

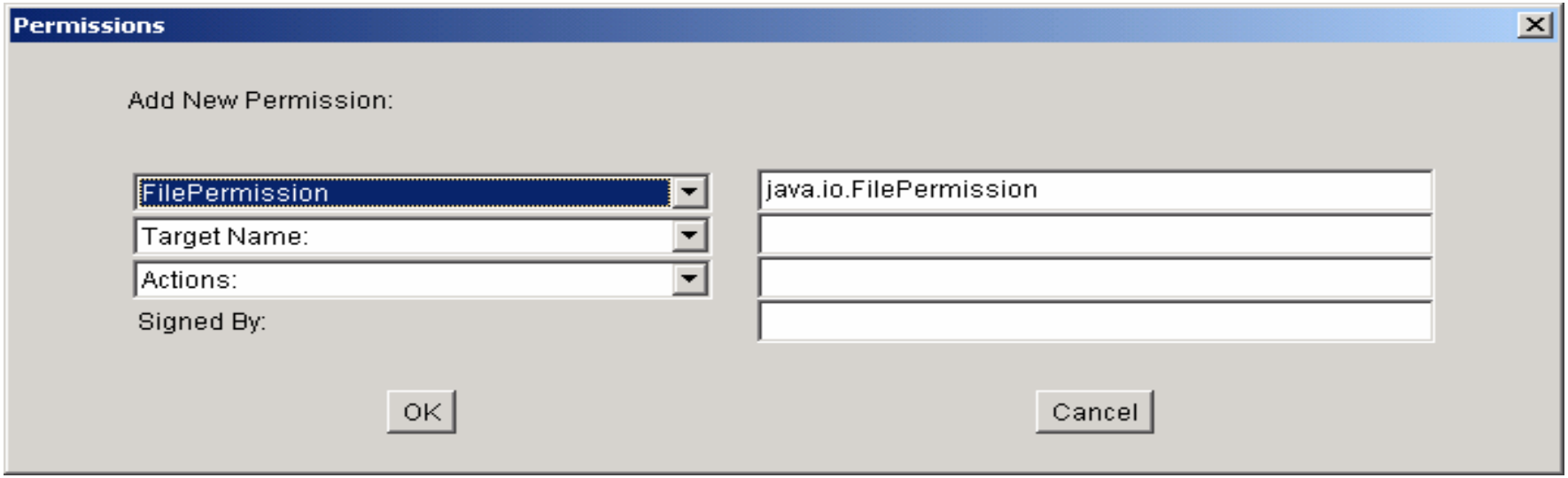


d) Po wypełnieniu pola CodeBase adresem URL i braku SignedBy należy przycisnąć przycisk **Add Permission** i wypełnić trzy pola pokazanego formularza

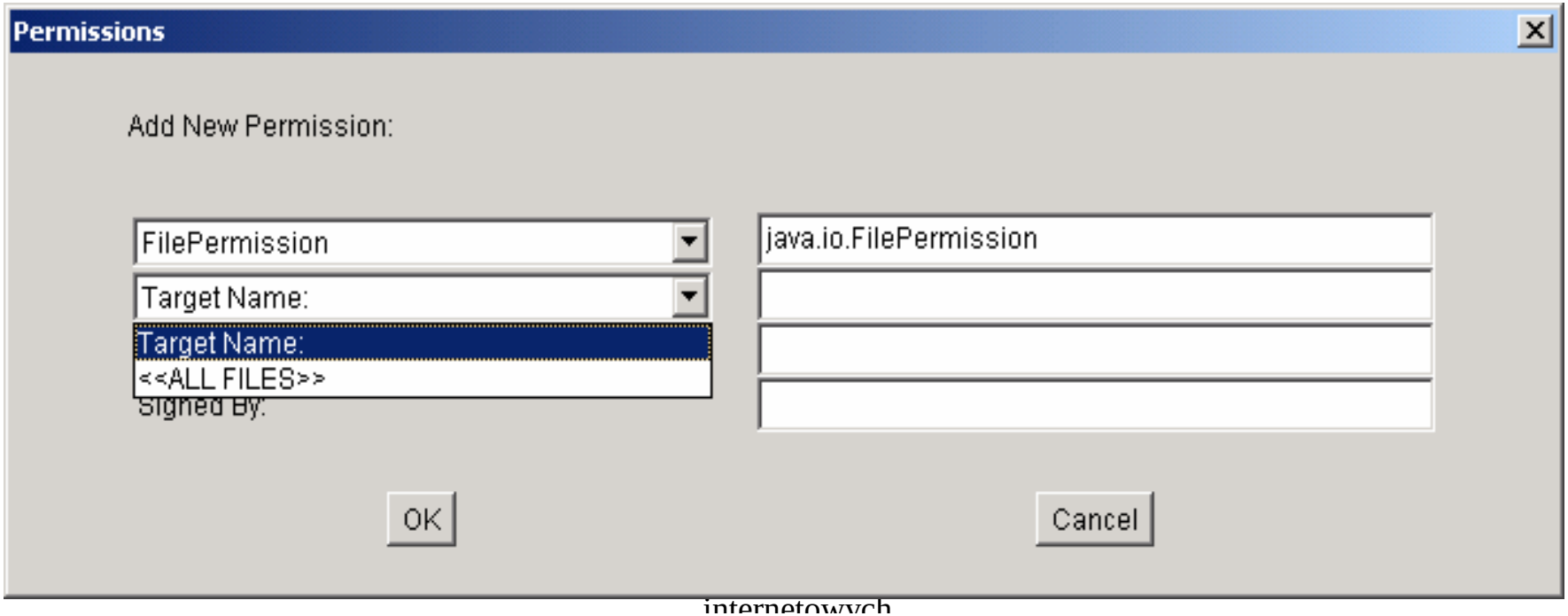




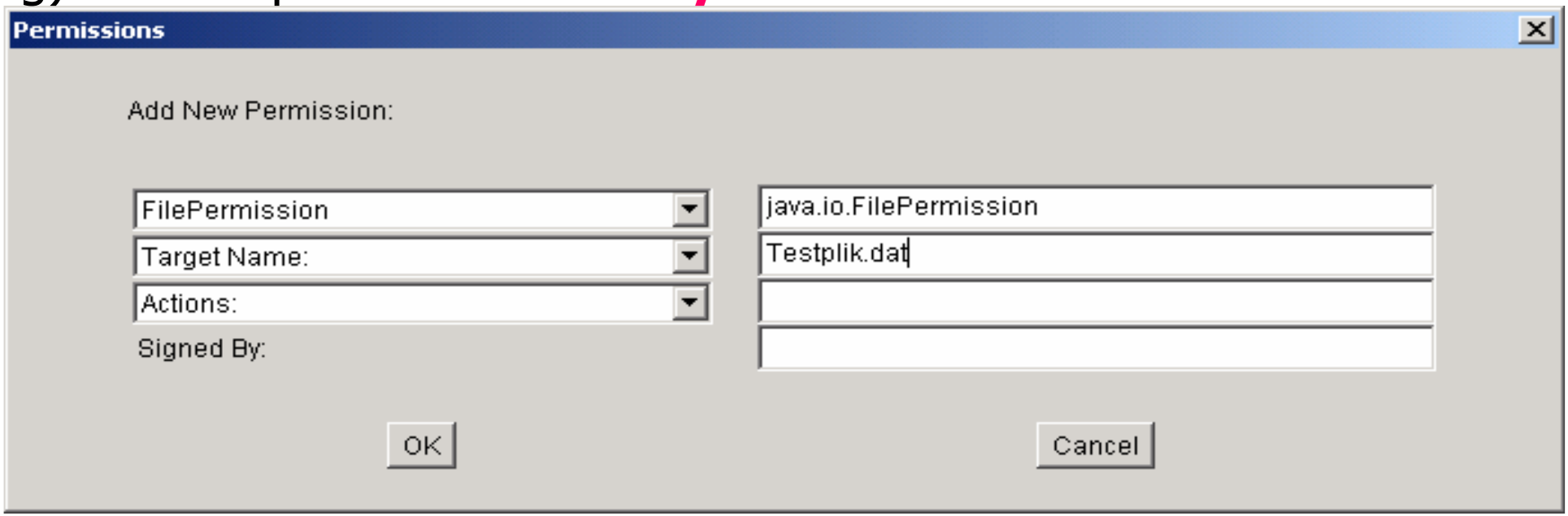
e) Wybrano pozwolenie na operacje plikowe



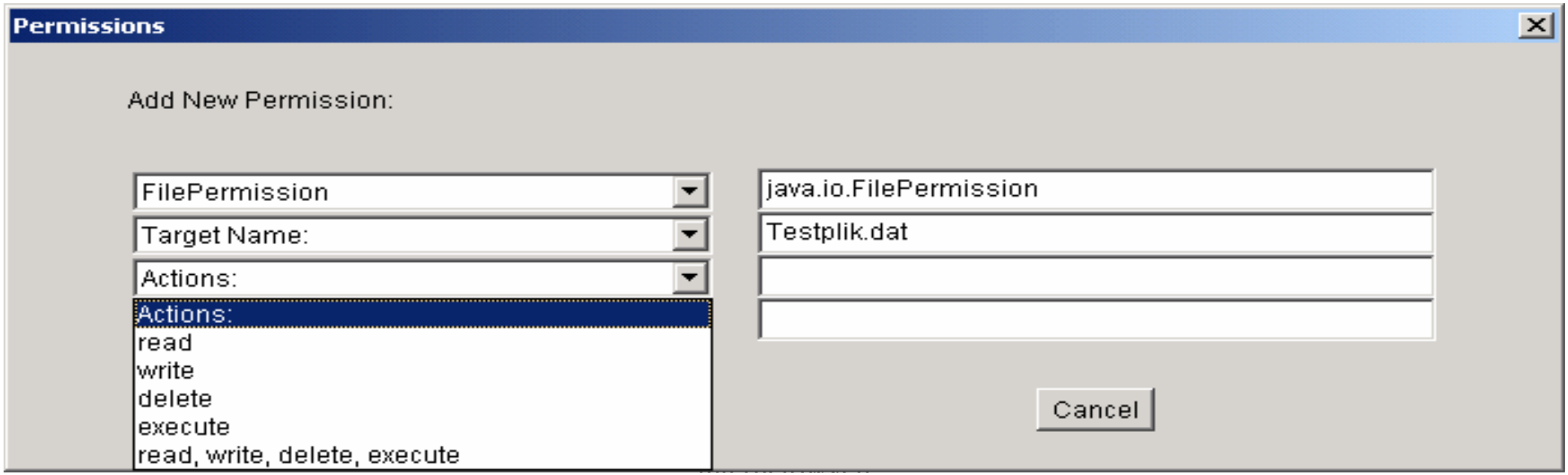
f) Wybór nazwy pliku do zapisu lub odczytu przez aplet



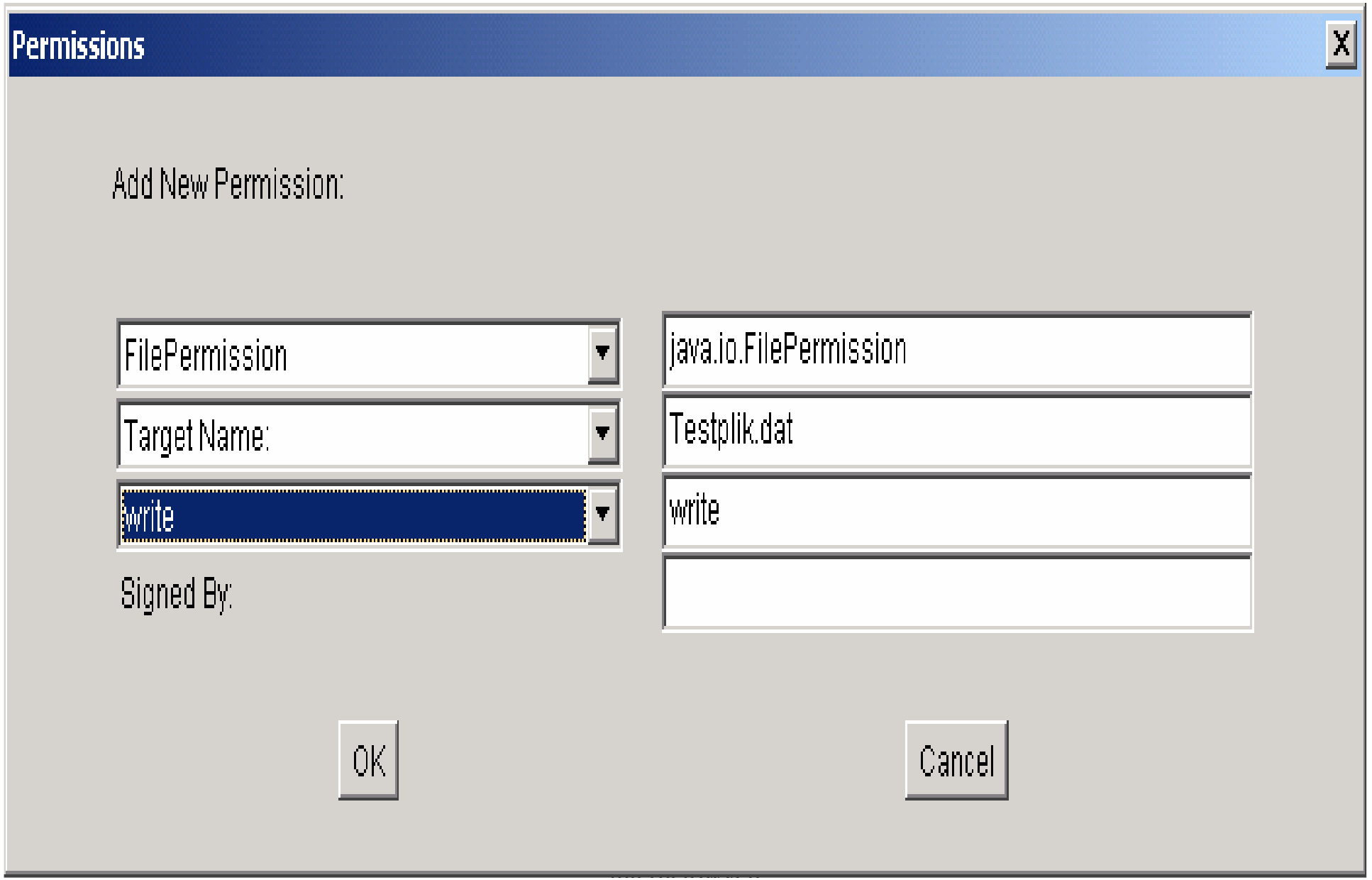
g) Podano plik o nazwie ***Testplik.dat***



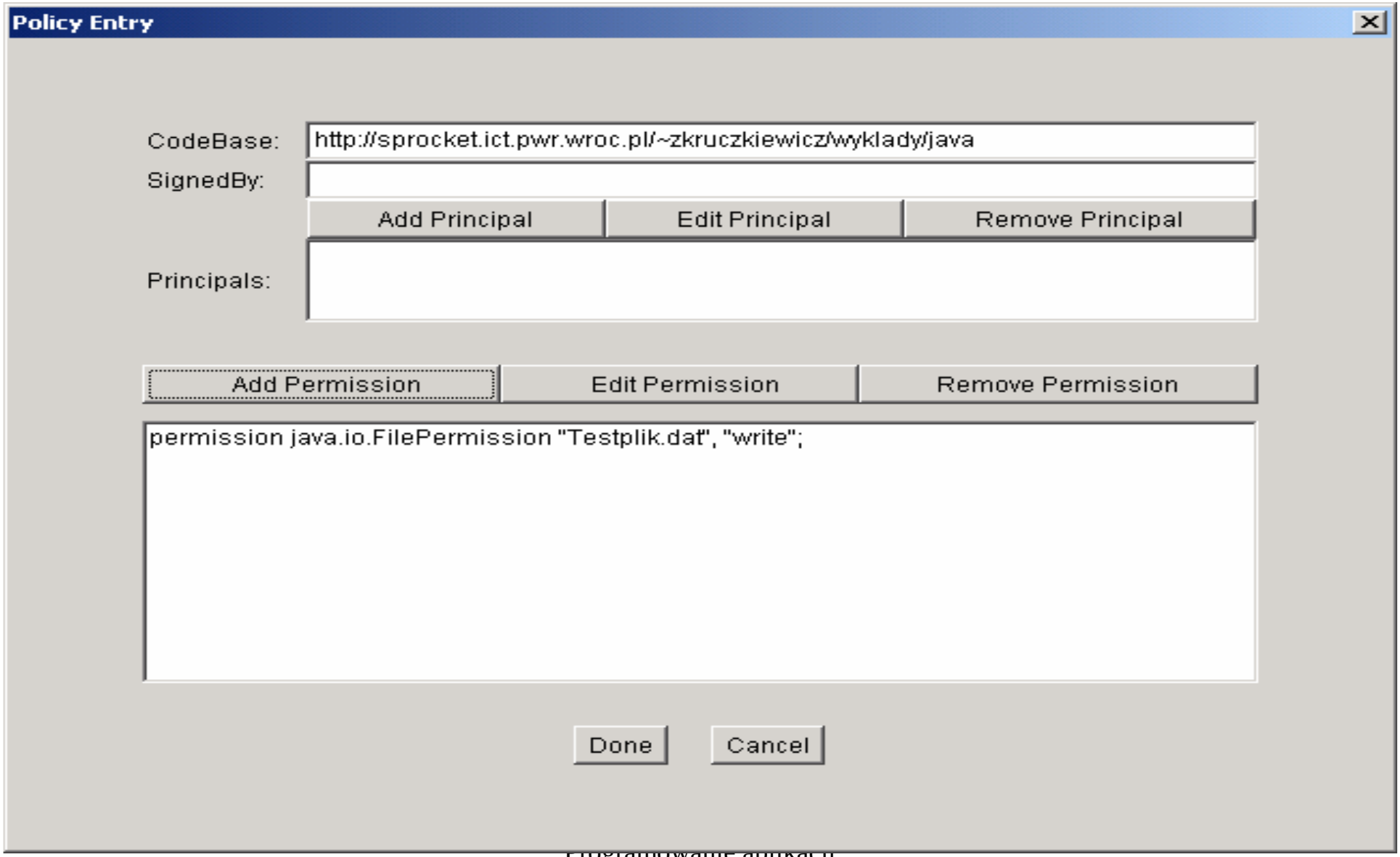
h) Wybór operacji np. zapisu do pliku ***Testplik.dat***



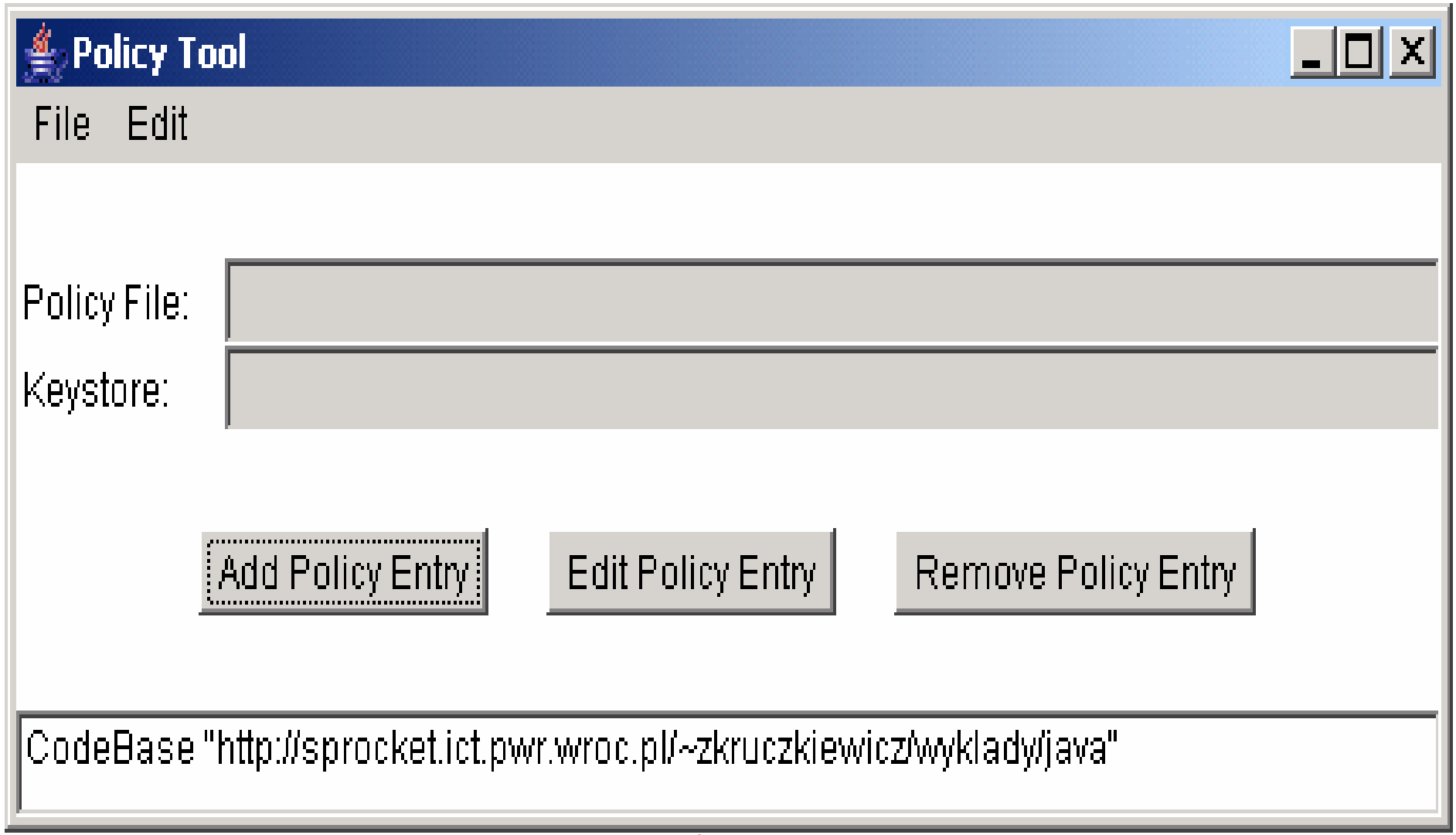
i) Pełna specyfikacja operacji wykonywanych przez aplet na komputerze przeglądarki. Należy nacisnąć klawisz **OK**



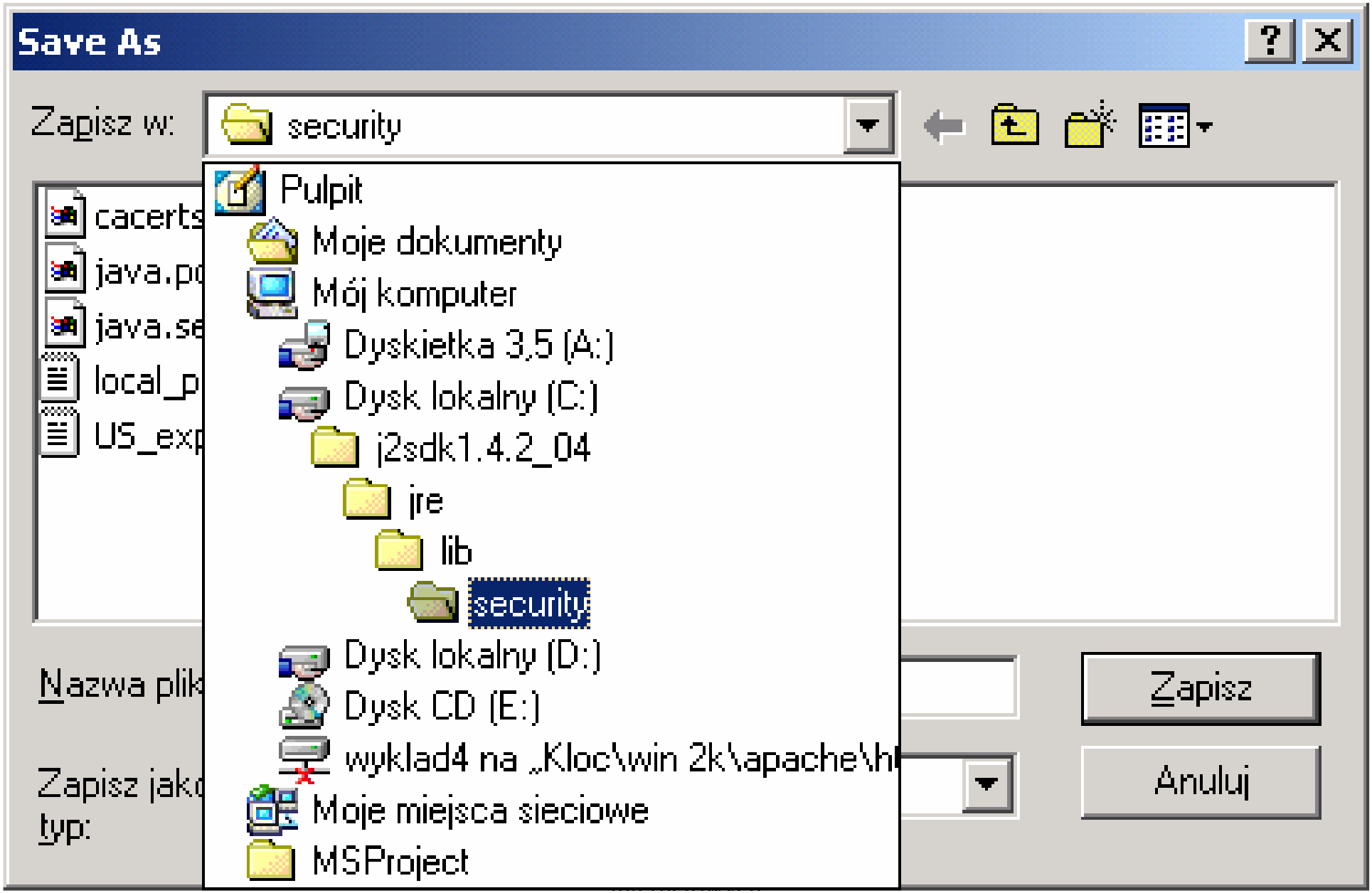
j) Po przejściu do kolejnego okna zatwierdzamy opcje bezpieczeństwa za pomocą klawisza **Done**



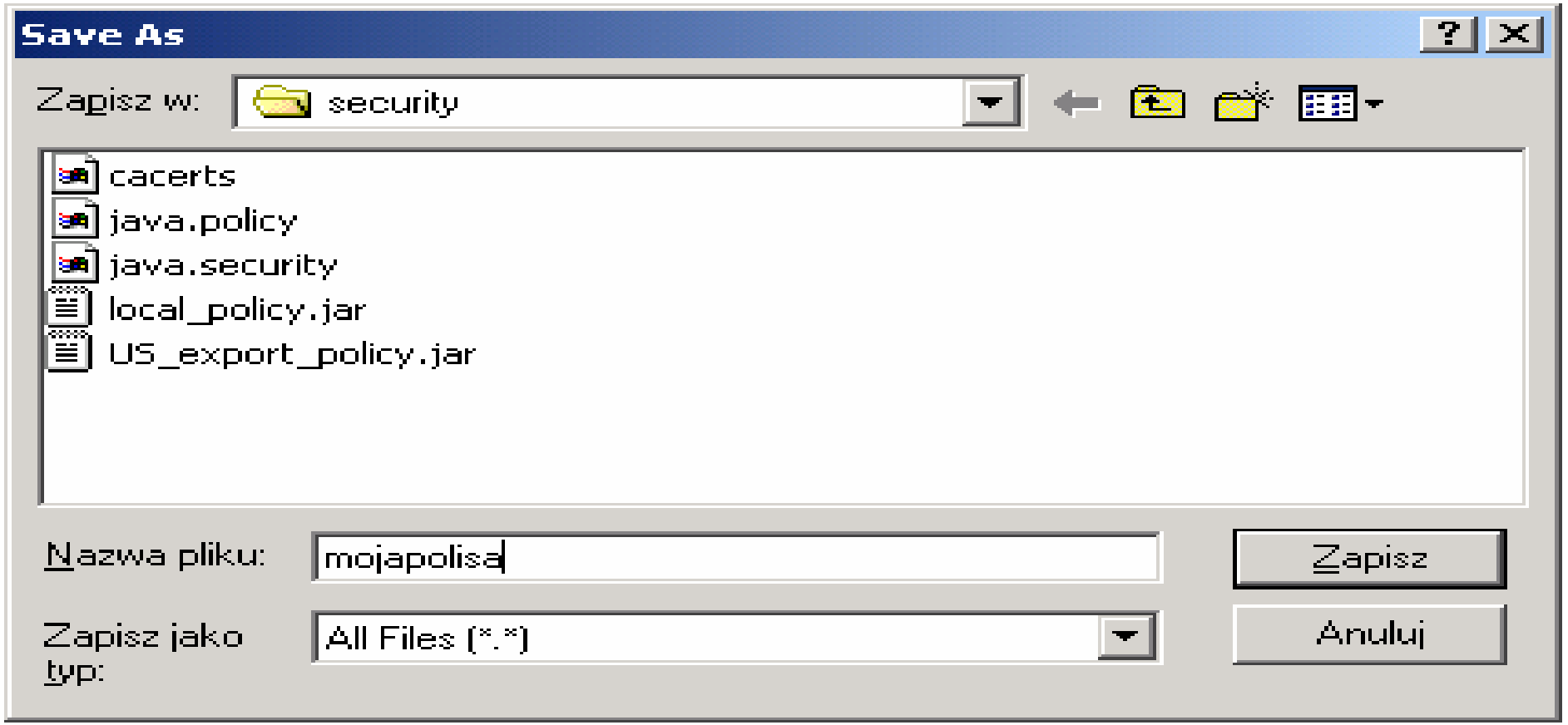
k) Należy teraz wybrać z opcji **File** opcję **Save As**



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



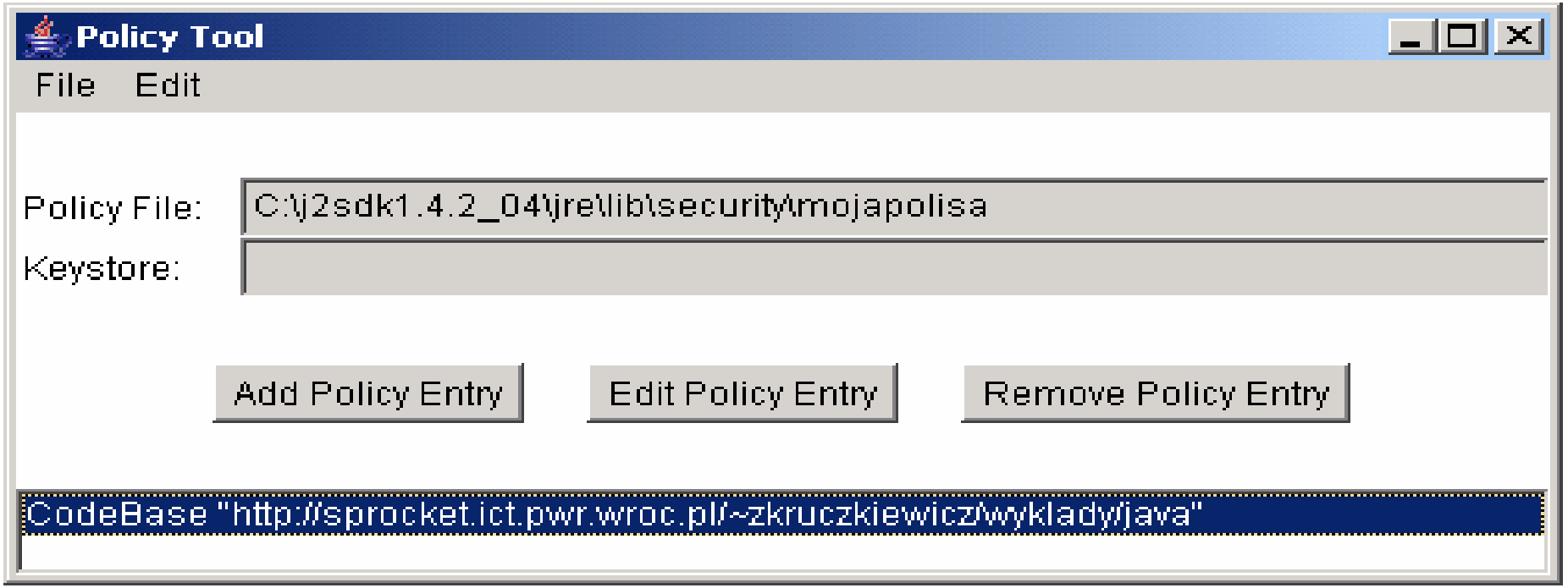
m) Należy zapisać plik uwierzytelniający po wypełnieniu pola tekstowego **Nazwa pliku** i zatwierdzeniu przyciskiem **Zapisz**



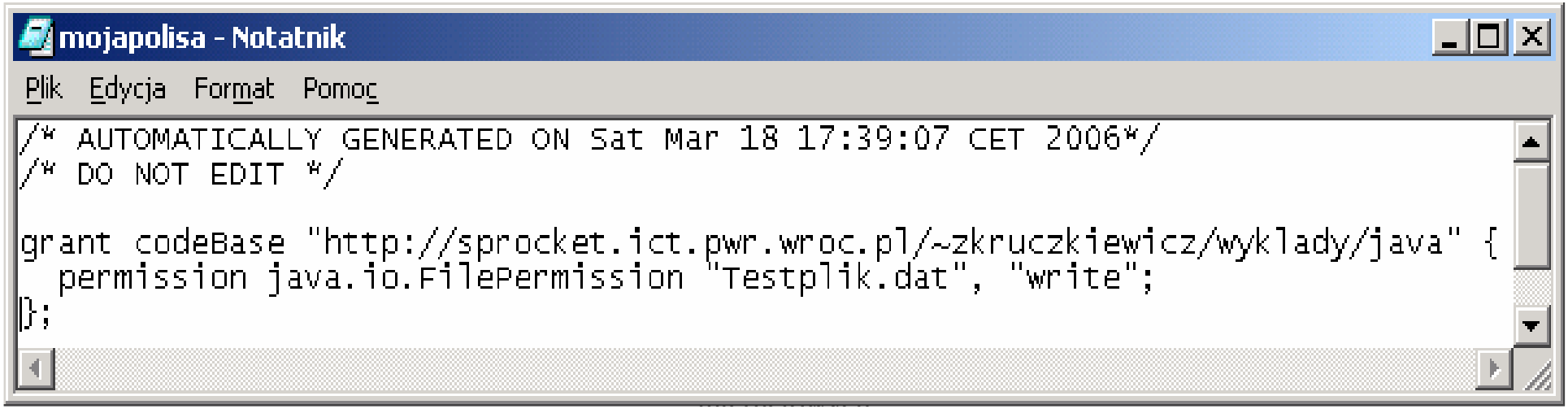
n) Należy potwierdzić zapis pliku uwierzytelniającego ***mojapolisa***

internetowych

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*** (jednia 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 uruchamoania 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
```

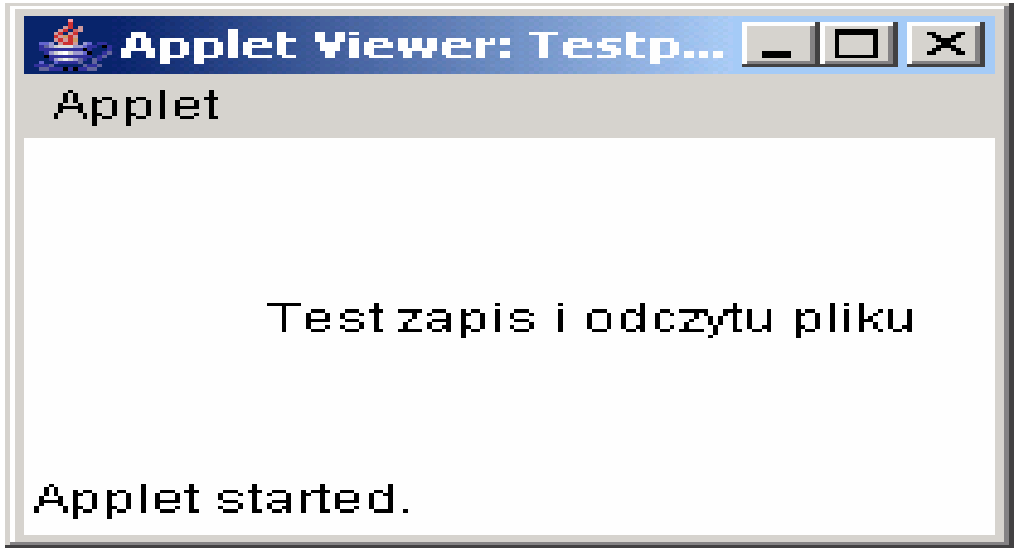
Przykład apletu realizującego czynności plikowe

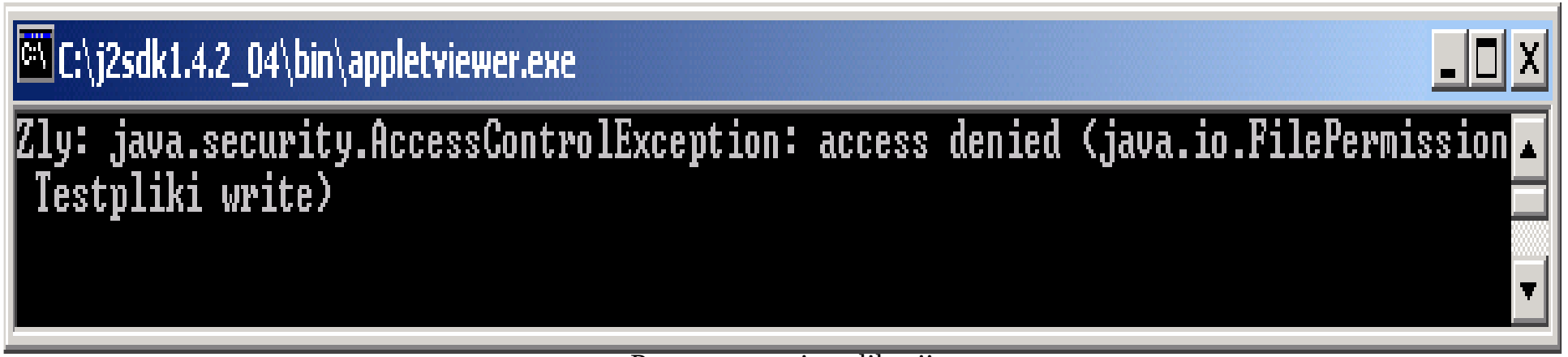
```
//próba zapisu i odczytu przez aplet Testpliki pliku o innej nazwie niż
//podano w pliku uwierzytelniającym, czyli Testplik zamiast
//Testplik.dat
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;

 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("Blad 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 ("Blad zapisu pliku bajtowego"+e);   }
   }
```

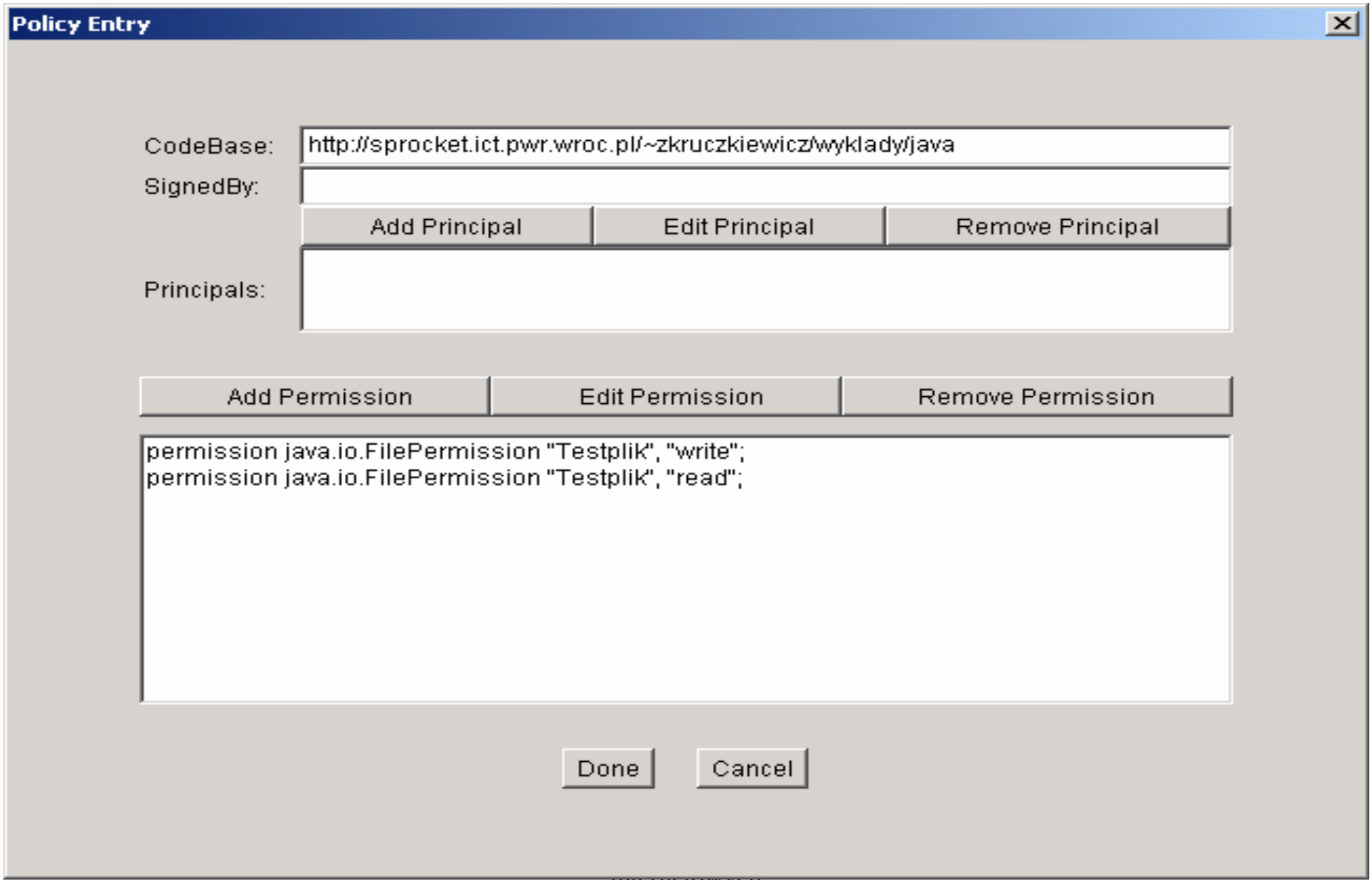
```
static void Odczytajplik2()
 { int dane=0;
   try
    {  FileInputStream plik = new FileInputStream ("Testplik");
       BufferedInputStream bufor = new BufferedInputStream (plik);
       dane=plik.read();
      while (dane!=-1)
        { System.out.print(dane);
          dane=bufor.read(); }
  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

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

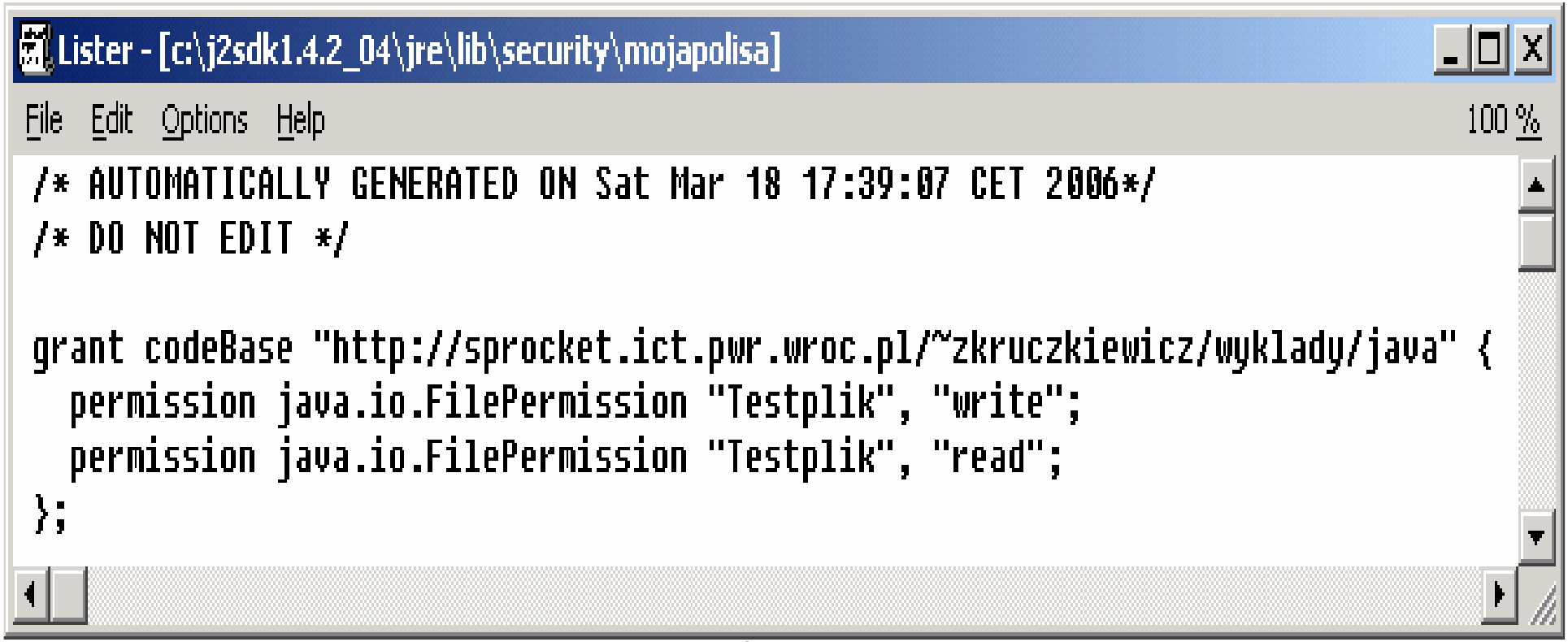




Po właściwym przygotowaniu pliku uwierzytelniającego ***mojapolisa***



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***.



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