Instytut Sterowania i Systemów Informatycznych
Uniwersytet Zielonogórski

Zaawansowane Systemy Decyzyjne

Laboratorium

prowadzący: Andrzej Czajkowski[1]

Dostęp do Bazy Danych.

1 Cel ćwiczenia.

Celem ćwiczenia jest wykorzystanie wbudowanego mechnizmu obsługi zewnętrznych appletów w celu pobrania danych przechowywanych w bazie danych.

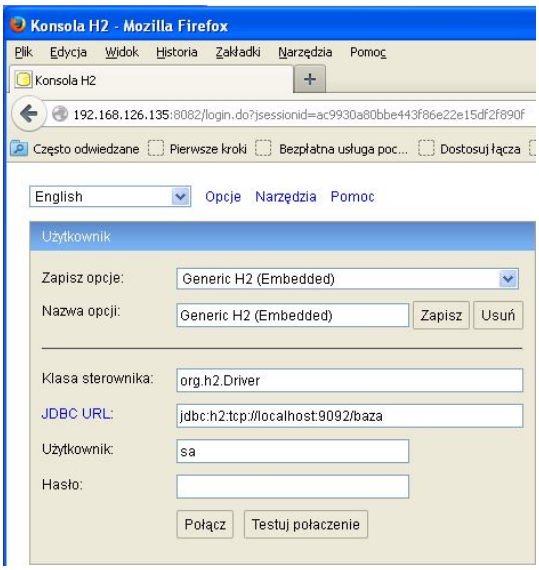
2 Wymagane oprogramowanie

1. Java Runtime Enviroment JRE w werssji 1.7u55 lub wyższej.
2. Corvid Exsys.
3. Eclipse Standard - `https://www.eclipse.org/downloads/`.
4. Baza Danych H2 - `http://www.h2database.com/html/main.html`. (W ćwiczeniu można wykorzystać dowolną bazę danych w dostępem za pomocą sterownika JDBC).
5. Apache Tomcat - `http://tomcat.apache.org/download-70.cgi`.

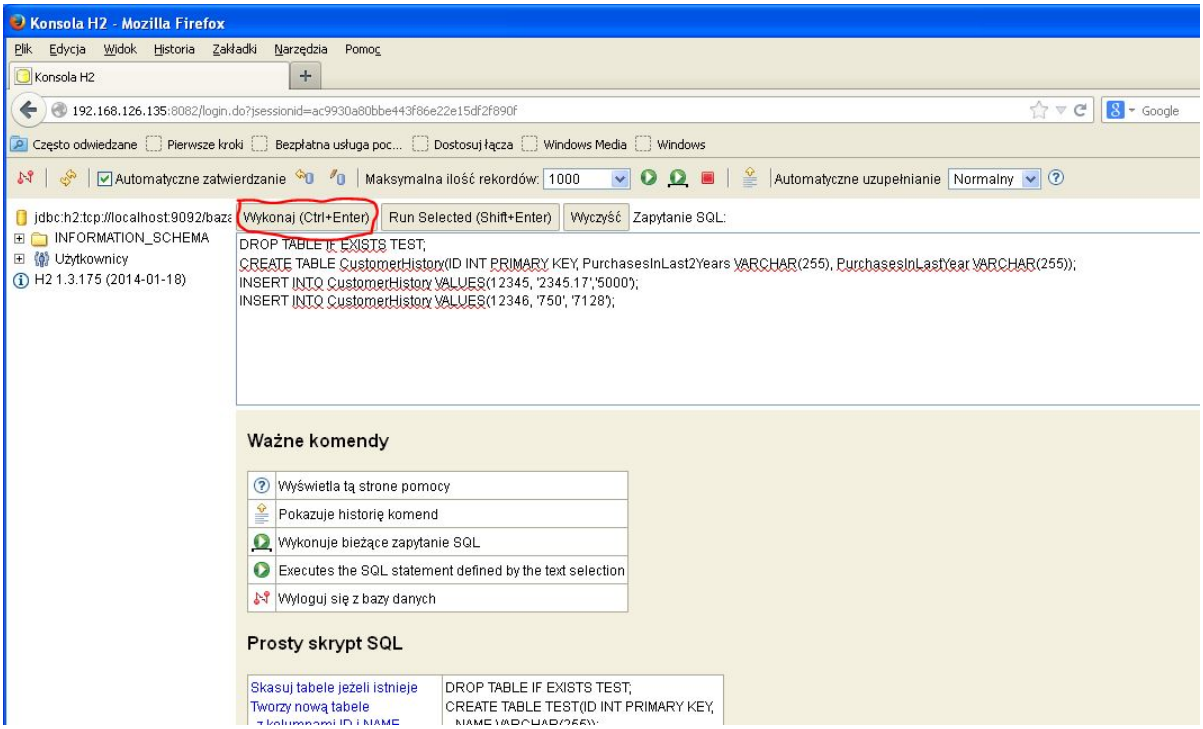
3 Przebieg ćwiczenia.

1. Konfiguracja servera Tomcat:
   - Zainstalowanie serwera z domyślnymi ustawieniami,
   - w folderze webapps (domyślnie: `C:/Program Files/Apache Software Foundation/Tomcat 7.0/webapps`) utworzyć folder `mojprojekt`.
2. Konfiguracja bazy danych H2:
   - Zainstalowanie bazy danych z domyślnymi ustawieniami,
   - Przekopiować plik `h2-1.3.175.jar` z folderu bazy danych (domyślnie: `C:/Program Files/H2/bin`) do folderu mojprojekt (w przypadku nowszej wersji bazy danych nazwa może się różnić, zmianę tą należy uwzględnić w dalszej części instrukcji),
   - W celu uruchomienia bazy z możliwością łączenia z appletu umieszczonego na stronie www należy utworzyć plik wsadowy w folderze mojprojekt z poleceniem: `java -jar h2-1.3.175.jar -web -webAllowOthers -tcp -tcpAllowOthers -browser`
   - uruchomić serwer bazy danych poprzez dwukrotne kliknięcie pliku wsadowego.
   - w narzędziu obsługi bazy danych należy wprowadzić domyślne parametry połączenia z bazą zgodnie z obrazkiem:

[1]Andrzej Czajkowski, Institute of Control and Computation Engineering, University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland. Email: `a.czajkowski@issi.uz.zgora.pl`



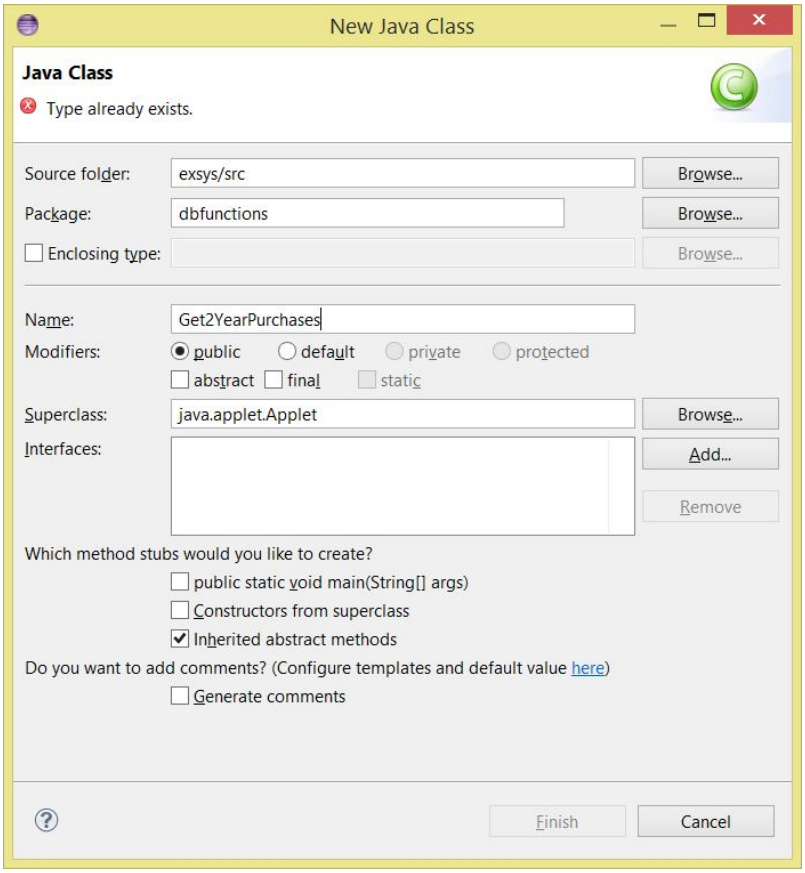
- w celu utworzenia odpowiedniej tabeli w bazie należy wykonać poniższe przykładowe instrukcje języka SQL:
  `DROP TABLE IF EXISTS TEST; CREATE TABLE CustomerHistory(ID INT PRIMARY KEY, PurchasesInLast2Years VARCHAR(255), PurchasesInLastYear VARCHAR(255)); INSERT INTO CustomerHistory VALUES(12345, '2345.17','5000'); INSERT INTO CustomerHistory VALUES(12346, '750', '7128');`
  W tym celu należy wprowadzić je zgodnie z poniższym obrazkiem i nacisnąć przycisk wykonaj:



3. Utworzenie appletu pośredniczącego pomiędzy systemem Exsys a bazą danych z wykorzystaniem środowiska programistycznego Eclipse:

   - Uruchomić środowisko Eclipse.
   - Utworzyć nowy Java Project o nazwie db_applet.

- W utworzonym projekcie utworzyć pakiet dbfunctions.
- W utworzonym pakiecie utworzyć klasę Get2YearPurchases dziedziczącą po klasie Applet.
  `public class Get2YearPurchases extends Applet`



- Należy dodać plik bibiloteki JDBC sterownika bazy H2 (menu Project->properties i właściwość java build path, następnie Add External JARs... i wyszukać plik h2-1.3.175.jar).
- Zaimplementować w klasie Get2YearPurchases wymianę informacji pomiędzy systemem Exsys a bazą danych:
  - W celu stworzenia poprawnej komunikacji należy utworzyć trzy pola w klasie:
    ```
    boolean running = false;
    String origName = "";
    Connection conn;
    ```
  - Następnie należy utworzyć metodę inicjalizującą applet:
    ```
    public void init() {
      if (running == false) {
         origName = this.getName();
         running = true;
      } else {
         processDataRequest(this.getName());
         this.setName(origName);
      }
    }
    ```

    Zadaniem tej metody jest przy pierwszym uruchomieniu (start systemu ekspertowego) zapisanie we wcześniej utworzonej zmiennej nadanej w pliku html nazwy dla appletu, za pomocą metody

`getName()` oraz zapamiętanie informacji o uruchomieniu appletu (zmienna logiczna `running`). Przy kolejnych wywołaniach metody `init()` (właściwe wywołanie w celu wyliczenia wartości w skutek przeprowadzanego wnioskowania) wykonywana jest część instrukcji warunkowej `else` i wywołanie metody przetwarzającej zapytanie, która utworzona będzie w kolejnym kroku. Parametry przesłane z systemu Exsys są pobierane za pomocą metody `getName()` i przesłane do metody przetwarzającej jako łańcuch znaków (**wymagane jest późniejsze parsowanie**). Na koniec w tym bloku następuje ponowne ustawienie nazwy appletu na odstawie nazwy zapamiętanej przy pierwszym uruchomieniu, w celu identyfikacji przez system Exsys (metoda `setName()`).

– W celu nawiązania połączenia z bazą danych należy utworzyć poniższą metodę:

```
public boolean OpenConn() {
   try {
     Class.forName("org.h2.Driver").newInstance();
     String url = "jdbc:h2:tcp://localhost:9092/baza";
     conn = DriverManager.getConnection(url, "sa", "");
     return true;
   } catch (Exception e) {
     e.printStackTrace();
     return false;
   }
}
```

– oraz metodę zamykającą połączenie:

```
public void CloseConn() {
  try {
     conn.close();
   } catch (SQLException e) {
     e.printStackTrace();
  }
}
```

– Metoda przetwarzająca zapytanie dla systemu Exsys:

```
void processDataRequest(String data) {
  if (OpenConn()) {
     String query = "SELECT PurchasesInLast2Years FROM CustomerHistory WHERE ID="+data;
     try {
       Statement st = conn.createStatement();
       ResultSet rs = st.executeQuery(query);
       while (rs.next()) {
         String s = rs.getString("PurchasesInLast2Years");
         returnDataToCorvid("" + s);
       }
     } catch (SQLException ex) {
       System.err.println(ex.getMessage());
     }
  }
}
```

Zadaniem tej metody jest wykonanie zapytania do bazy danych. W tym celu wpierw należy otworzyć połączenie z bazą za pomocą metody `OpenConn()`. W przypadku błędu połączenia zostanie wyświetlony stosowny komunikat na konsoli JAVA. Następnie należy przygotować zapytanie w języku SQL na podstawie danych otrzymanych z systemu Exsys (wartość w tym przypadku odnosi się do numeru klienta i jest przechowywana w zmiennej `data`). Następnie należy wykonać zapytanie za pomocą

metody `exeuteQuery()` i zwrócony wynik przesłać z powrotem do systemu Exsys (w przypadku zapytania które zwróci kilka wierszy danych z wykorzystaniem pętli `while` wszystkie zostaną przesłane do systemu Exsys).

– Metoda zwracająca wartości do systemu Exsys na podstawie przetworzonego wcześniej zapytania:

```
public void returnDataToCorvid(String resultData) {
   Applet corvidRuntime;
   corvidRuntime = getAppletContext().getApplet("CorvidRuntime");
   corvidRuntime.setName(resultData);
   corvidRuntime.start();
}
```

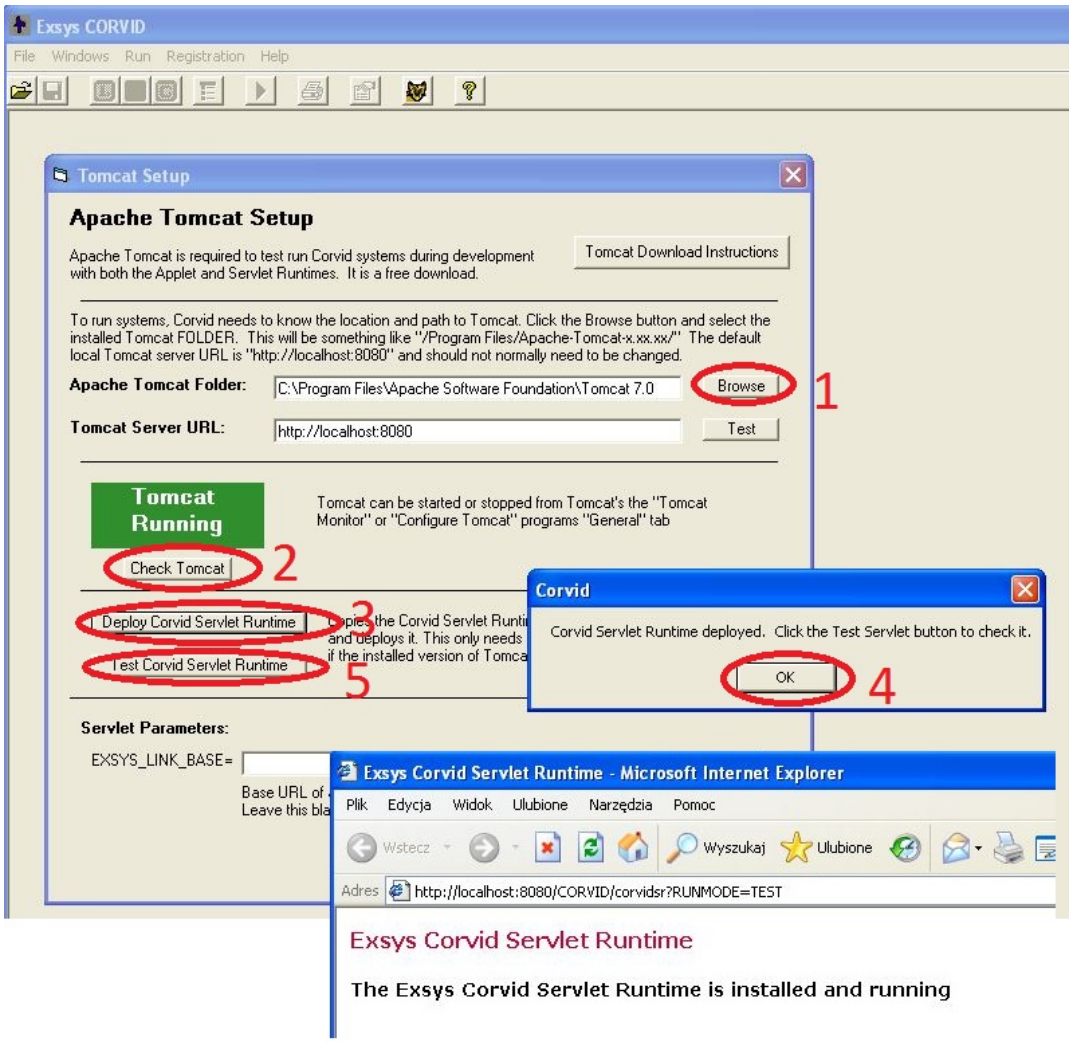
Zadaniem tej metody jest wykorzystanie metody `setName()` do przesłania uzyskanych w skutek działania appletu wyników do appletu Exsysa (systemy ekspertowe w systemie Exsys są kompilowane do appletów i uruchamiane we wbudowanej lub zewnętrznej przeglądarce). W metodzie tej lokalizowany jest po nazwie applet Exsys'a i wysyłane są do niego dane przesłane z metody przetwarzającej za pomocą zmiennej resultData będącej argumentem tej metody.

– Ostatnim krokiem w Eclipsie jest wyeksportowanie utworzonej klasy jako archiwum JAR. Aby to wykonać należy uruchomić eksportera plików poprzez menu File->Export. Następnie wybrać żądany typ czyli JAR file z folderu Java. W kolejnym kroku należy wyszukać i zaznaczyć klasy, które mają zostać wyeksportowane (w przypadku tej instrukcji można zaznaczyć cały projekt). Wprowadzić nazwę archiwum (np. dbapplet) i kliknąć `finish` (na potrzeby tej instrukcji nie trzeba zaznaczać nic więcej). Ze względu na implementację interfejsu serializacji przez klasę Applet możliwe jest wyświetlenie o ostrzeżeniach w projekcie (Warning) w trakcie eksportowania. W tym przypadku można je zignorować.

– uzyskany plik appletu należy umieścić w folderze mojprojekt

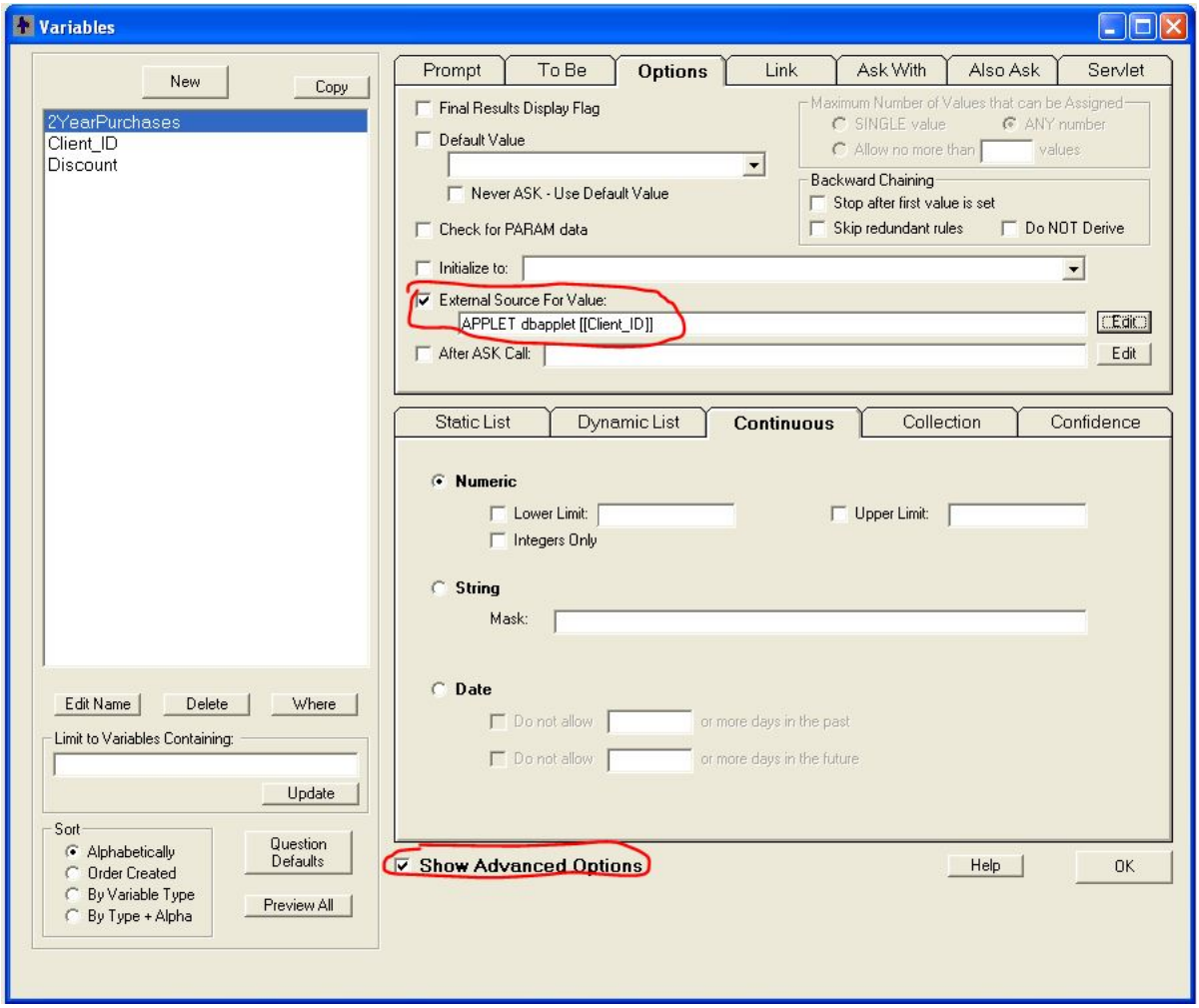
4. konfiguracja serwera tomcat w środowisku Tomcat:

- w celu konfiguracji serwera należy wykonać instrukcje:
  1 Wprowadzić ścieżkę instalacyjną Tomcat'a.
  2 Sprawdzić czy serwer jest uruchomiony ().
  3 Zainstalować na serwerze serwlet systemu Exsys.
  4 Potwierdzić komunikat instalacyjny serwletu.
  5 Sprawdzić jego funkcjonowanie.

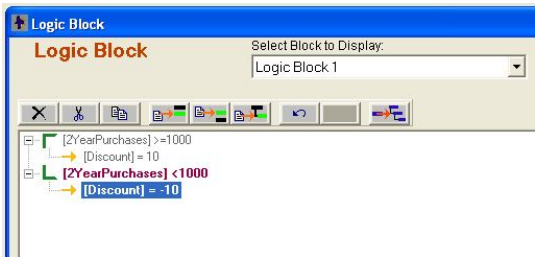


5. Wykorzystanie danych pochodzących z bazy danych w systemie Exsys:

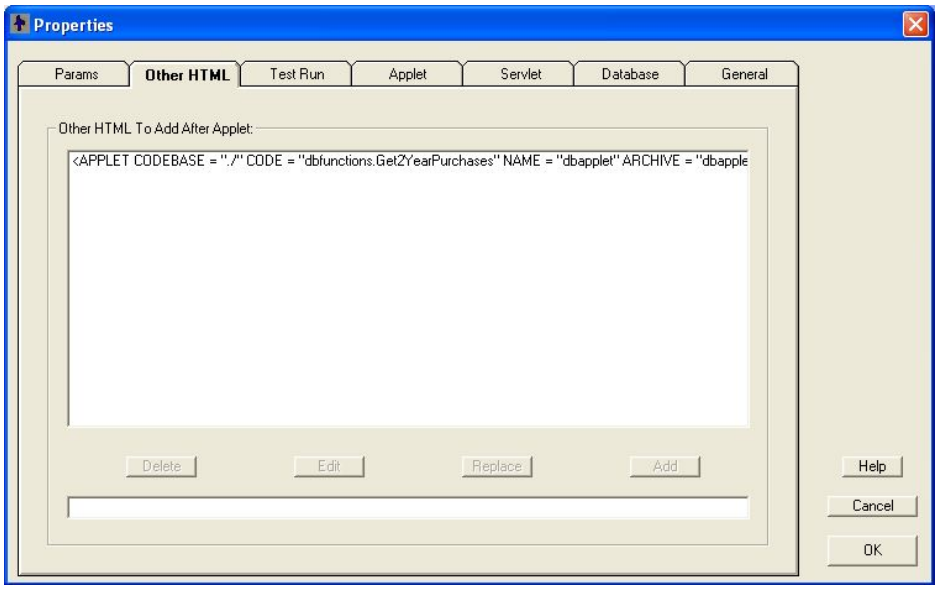
   - W folderze mojprojekt utworzyć nowy projekt np. dbprojekt.
   - dodać trzy zmienne:
     (a) 2YearPurchases- zmienna typu numeric
     (b) Client_ID- zmienna typu numeric
     (c) Discount - zmienna typu confidence
   - W celu uzyskania wartości dla zmiennej 2YearPurchases wprowadzamy pobranie danych z bazy z wykorzystaniem napisanego wcześniej appletu zgodnie z poniższym rysunkiem:



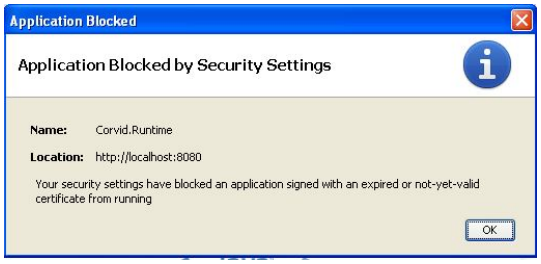
- Należy zwrócić uwagę na poprawne wprowadzenie nazw appletu i zmiennej z numerem klienta.
- Kolejnym krokiem jest zbudowanie bloku logicznego zgodnie z regułą aby klienci którzy wydali ponad 1000 uzyskiwali rabat a w przeciwnym wypadku nie. Taką logikę obrazuje rysunek:



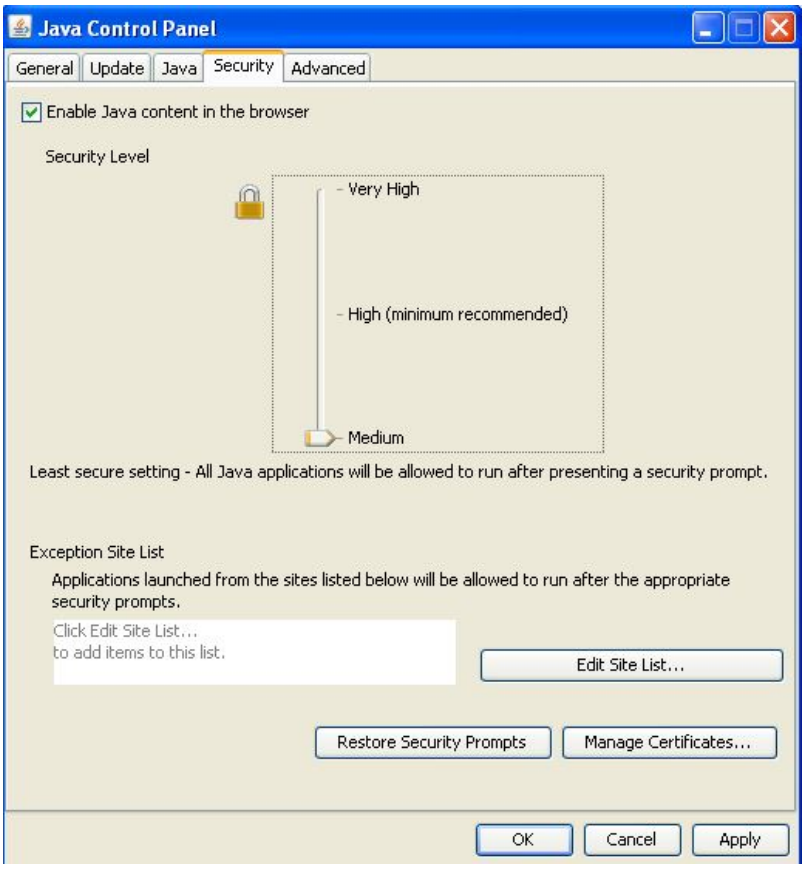
- Ostatnim krokiem jest wprowadzenie modyfikacji do pliku html z lokalizacją potrzebnych plików poprzez dodanie polecenia:
  `<APPLET CODEBASE = "./" CODE = "dbfunctions.Get2YearPurchases" NAME = "dbapplet" ARCHIVE = "dbapplet.jar, h2-1.3.175.jar" WIDTH = 0 HEIGHT = 0 H SPACE = 0 VSPACE = 0></APPLET>`
  w zakładce `Other HTML` która znajduje się w oknie `properties`.



6. W przypadku komunikatu o błędzie certyfikatu appletu CorvidExsys:



Należy zmienić ustawienia bezpieczeństwa w celu odpytywania użytkownika w przypadku uruchamiania potencjalnie niebezpiecznych apletów zgodnie z poniższym rysunkiem (opcję konfiguracji JRE można znaleźć w panelu sterowania):


Java Control Panel
General
Update
Java
Security
Advanced
Enable Java content in the browser
Security Level
Very High
High (minimum recommended)
Medium
Least secure setting - All Java applications will be allowed to run after presenting a security prompt.
Exception Site List
Applications launched from the sites listed below will be allowed to run after the appropriate security prompts.
Click Edit Site List...
to add items to this list.
Edit Site List...
Restore Security Prompts
Manage Certificates...
OK
Cancel
Apply