Programowanie obiektowe

Java

Autor: dr inż. Zofia Kruczkiewicz

Literatura:

- **L. Lemay, R. Cadenhead** Java 2 dla każdego
- **P. Naughton** Podręcznik Języka Programowania Java
- **Krzysztof Barteczko** JAVA, wykłady i ćwiczenia
- **Boone Barry** Java dla programistów C i C++

Linki do oprogramowania Java SDK1.4.2

Java 2 Software Development Kit, Standard Edition, version 1.4.2 –

http://java.sun.com/j2se/1.4.2/download.html

Środowiska programistyczne np:

- Borland Jbuilder
- Symantec Visual Café
- JCreator http://www.jcreator.com/download.htm
  (wersja free http://www.jcreator.com/download.htm)
- Eclipse http://www.eclipse.org/downloads/index.php
- Help (http://www.allimant.org/javadoc/jdk14e.html)

1. Charakterystyka języka Java

1) **Obiektowy język Java** – składnia języka podobna do języka C++.

   Pliki źródłowe: nazwa_klasy_publicznej.java,

   gdzie nazwa musi być nazwą klasy publicznej, zdefiniowanej w tym pliku.

2) **Kompilator** przetwarza program nazwa_klasy_publicznej.java na kod binarny zwany B-kod (bytecode, J-code)

   Pliki po kompilacji: nazwa_klasy_publicznej.class

   B-kod może być zinterpretowana i wykonywana przez maszynę wirtualną Java (JVM Java Virtual Machine), czyli urządzenie logiczne

3) **Maszyna wirtualna Java** (JVM Java Virtual Machine).

   JVM jest abstrakcyjnym komputerem, który wykonuje programy nazwa.class:

   a. interpretator wbudowany w przeglądarkę WWW,

   b. oddzielny program

   c. Just-In-Time (przetworzenie nazwa.class na program wykonalny specyficzny dla danej maszyny)

4) **Biblioteka Javy** – pakiety z oprogramowaniem wspomagającym tworzenie programów działających w sieci np. Internet, umożliwiającym tworzenie interfejsu użytkownika, ogólnego przeznaczenia

2. Tworzenie programu w Javie

- Aplikacja (application) - program interpretujący aplikacje jest uruchamiany w systemie operacyjnym (java.exe)

  Program zawierający między innymi jeden moduł źródłowy, którego klasa publiczna zawiera publiczną metodę klasową o nagłówku

  public static void main(String args[])

- Aplet (applet) – program interpretujący aplety jest wbudowany np. w przeglądarkę www

  Program zawierający między innymi jeden moduł źródłowy, którego klasa publiczna zawiera miedzy innymi podstawowe metody: init(), start(), stop(), paint(), destroy()

Uwaga: możliwe jest napisanie programu w Javie, który będzie pracował jako applet i jako aplikacja.

2.1. Tekst źródłowy w Javie

```
public class Witaj
{
 public static void main(String args[])
 {
    System.out.print("Dzien dobry, nazywam się Jan Kowalski\n");
 }
}
```

2.2. Kompilacja

javac Witaj.java

gdzie położenie (katalog) programu javac (kompilator Javy) powinno być znane systemowi operacyjnemu, a katalog bieżący powinien zawierać plik źródłowy Witaj.java. Zostanie wygenerowany plik Witaj.class z instrukcjami dla JVM

2.3. Interpretacja

java Witaj

Interpretator java (położenie znane systemowi operacyjnemu)

- wyszuka plik o nazwie *Witaj.class* w katalogu bieżącym
- sprawdzi, czy klasa *Witaj* posiada publiczną metodę statyczną *main*
- wykona instrukcje zawarte w bloku funkcji *main*, czyli wyświetli na ekranie napis

  Dzien dobry, nazywam się Jan Kowalski

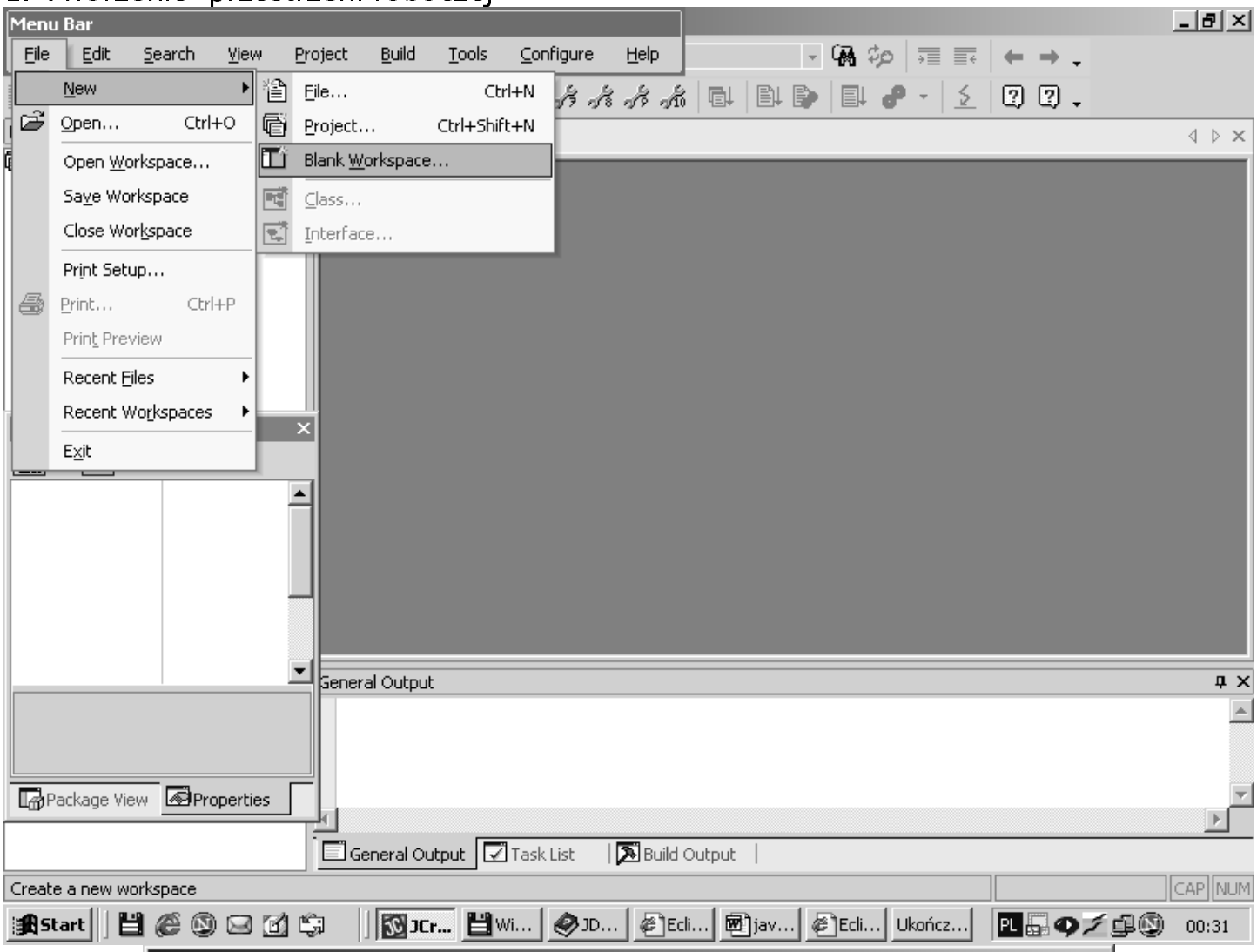
  i przejdzie do następnego wiersza

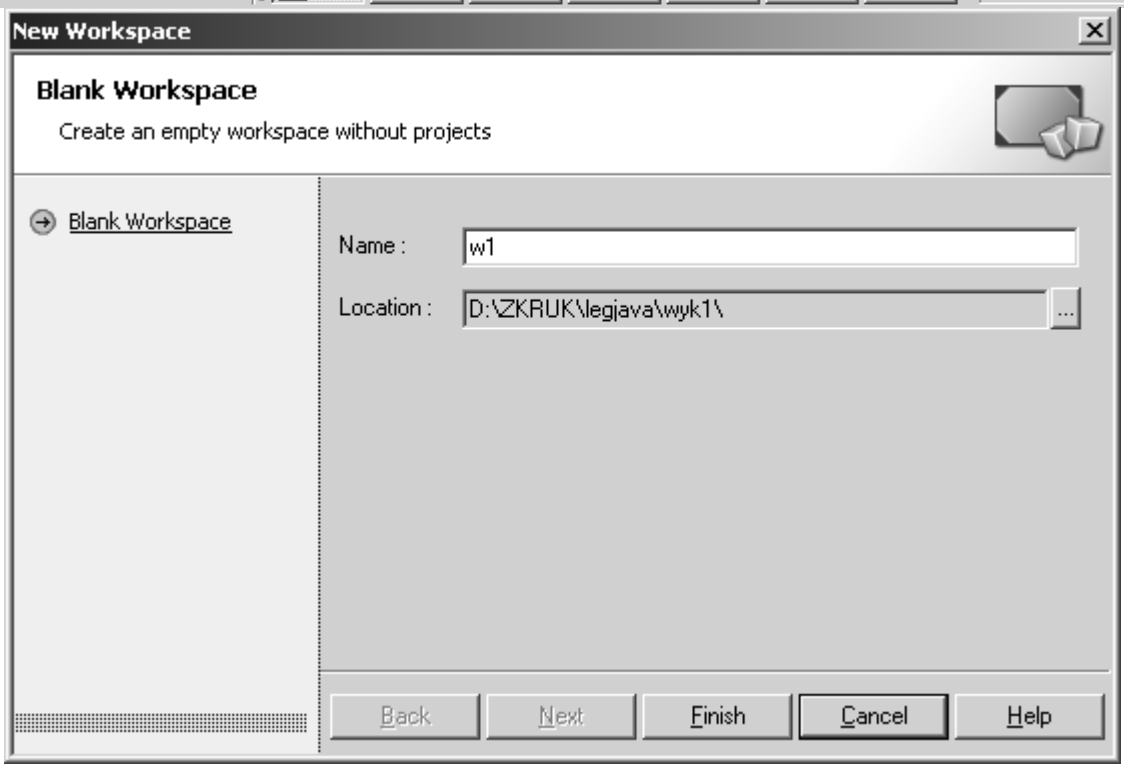
Uwagi:

- do metody main z wiersza rozkazowego jako parametr jest przekazywana tablica args obiektów (łańcuchów) klasy *String* - w klasie Witaj jest ona pomijana
- każda instrukcja kończy się średnikiem

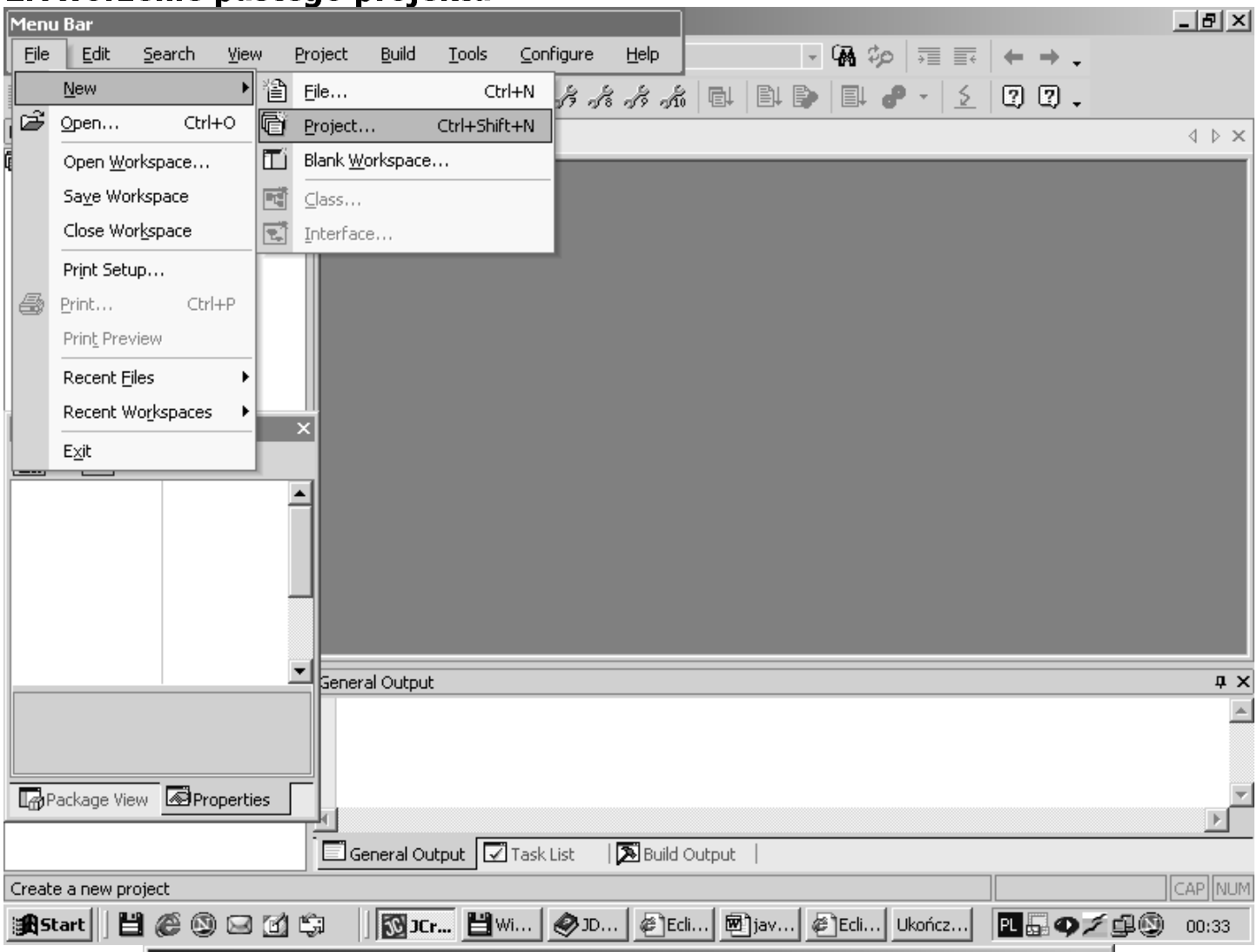
Środowisko JCreator

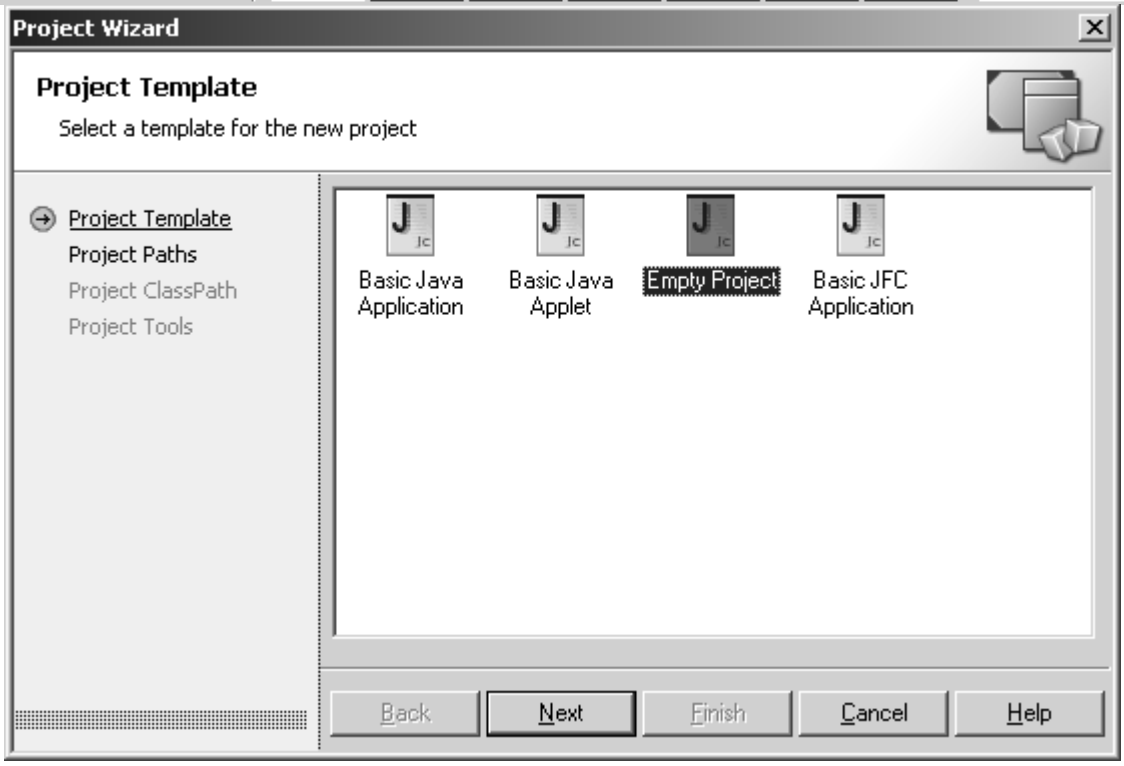
1. Tworzenie przestrzeni roboczej

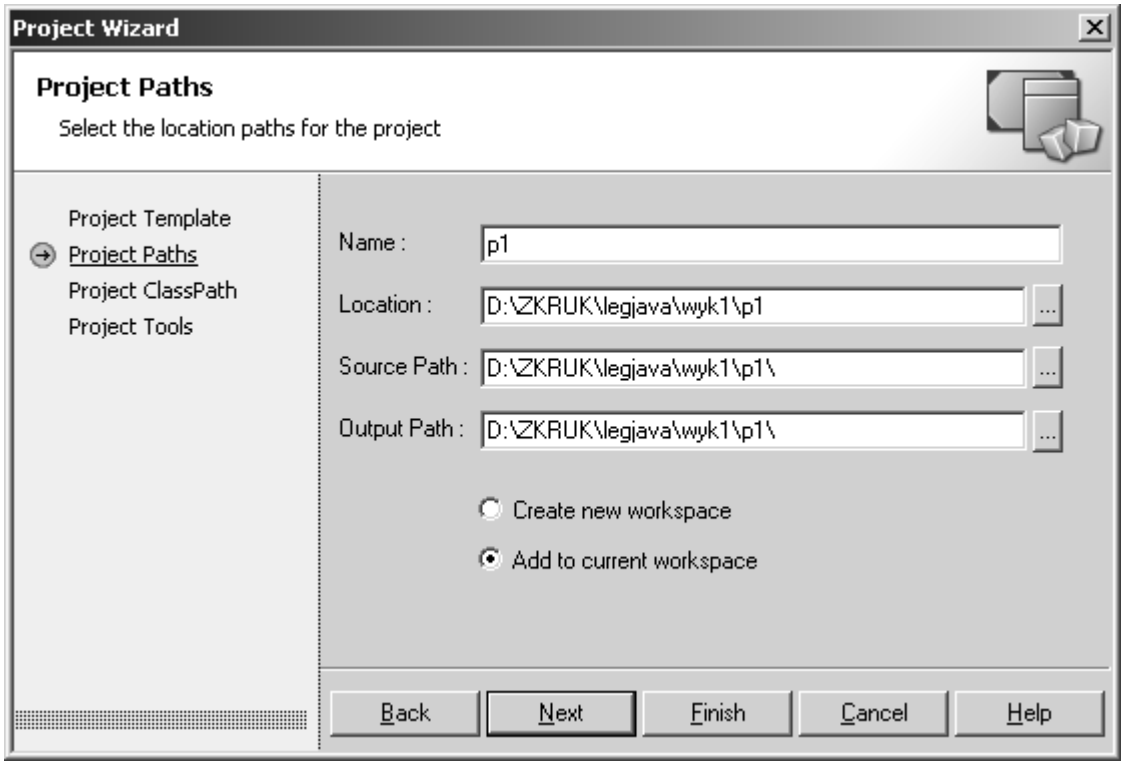




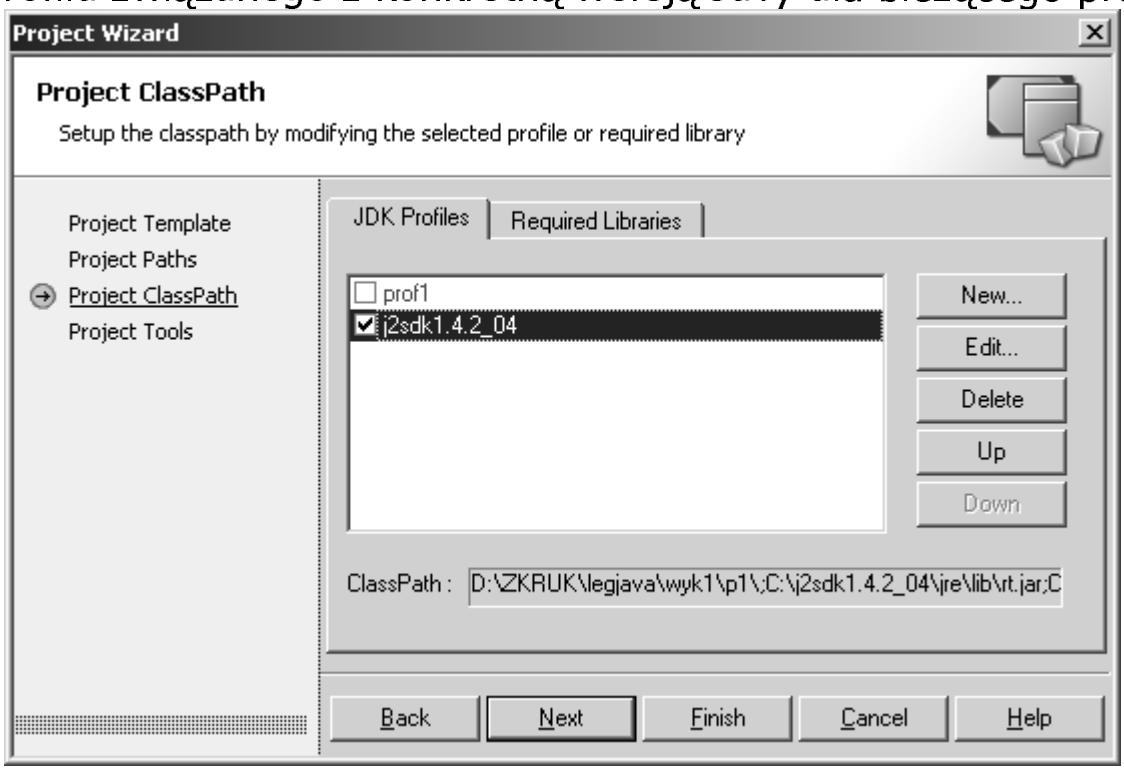
2.Tworzenie pustego projektu



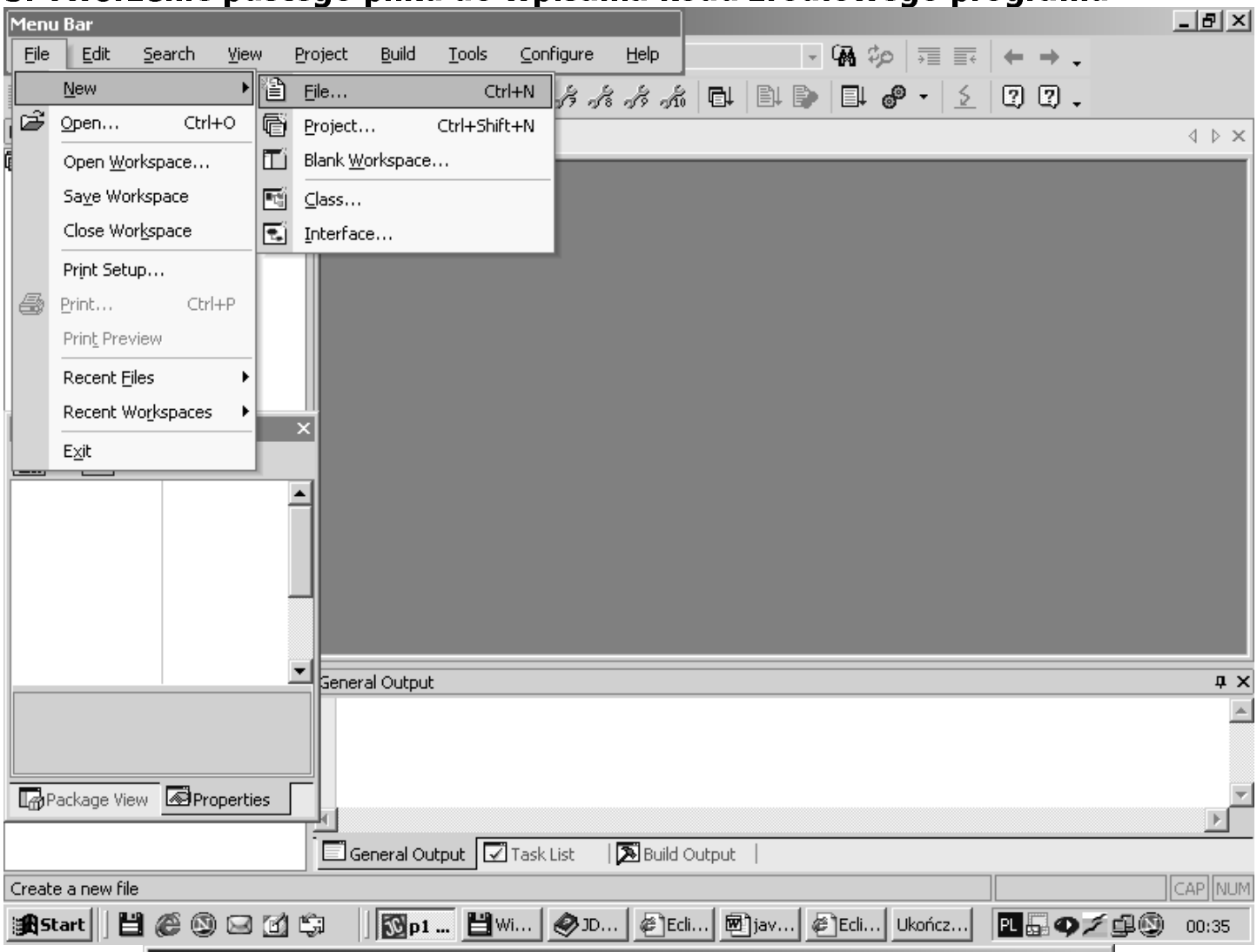


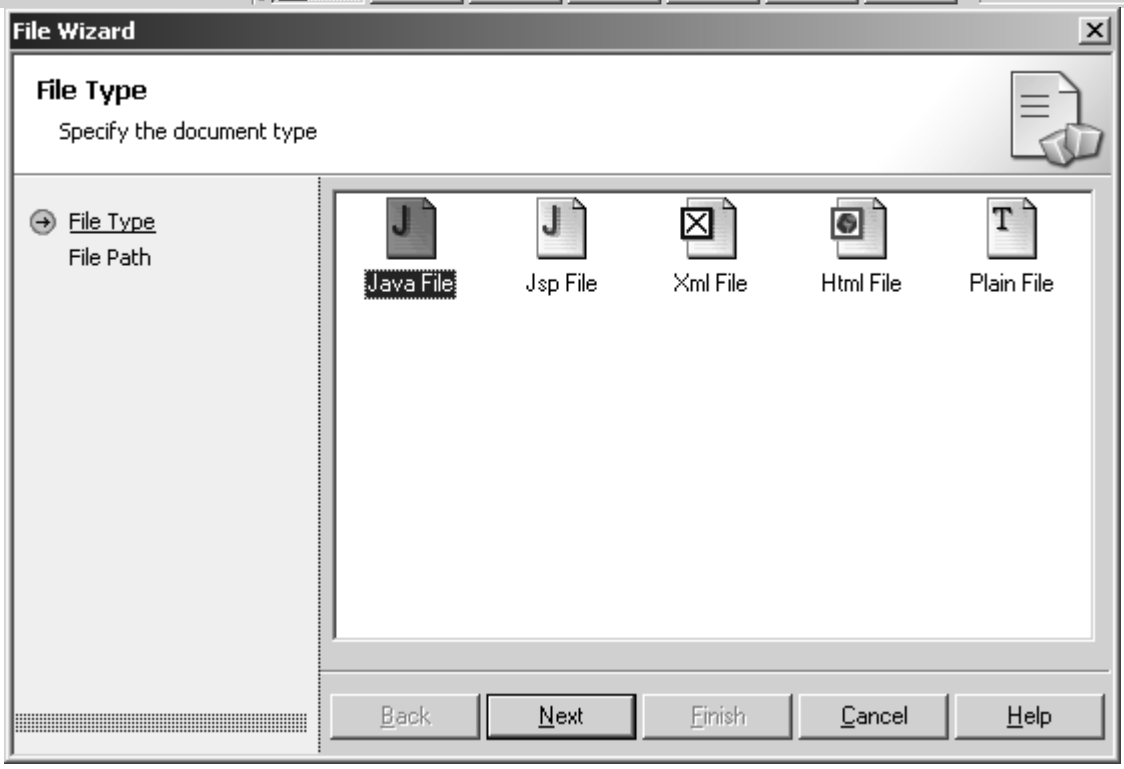


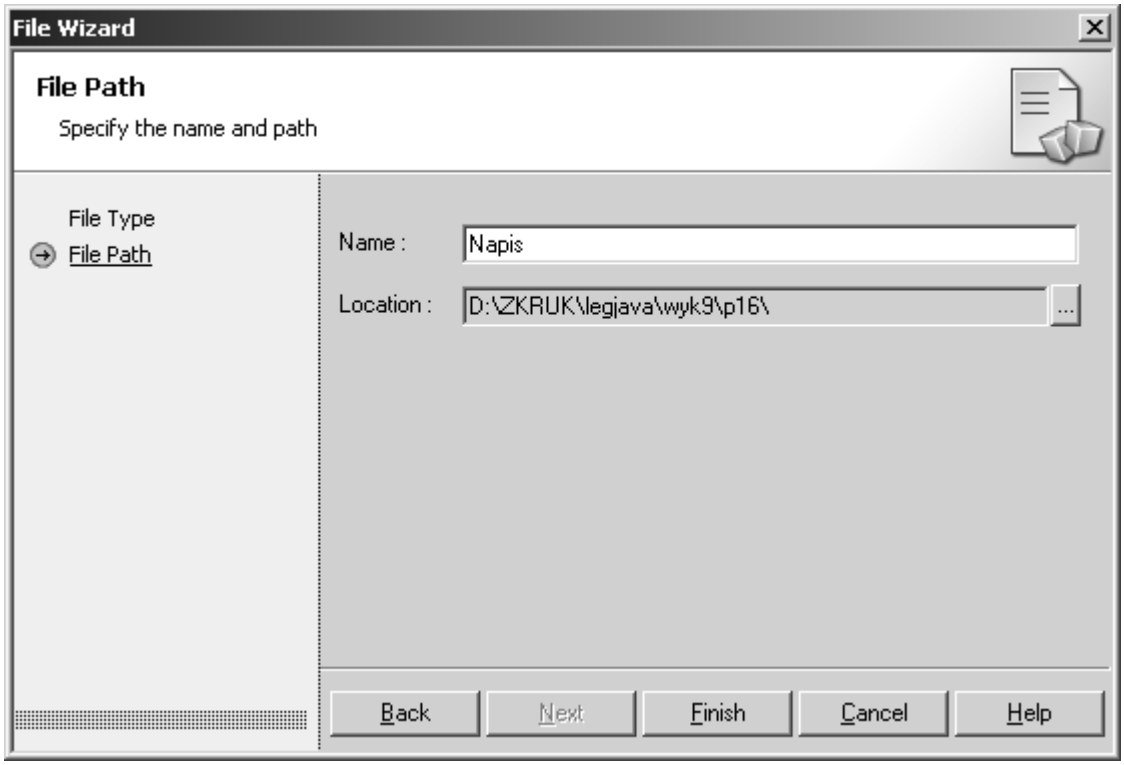
Wybór profilu związanego z konkretną wersją Javy dla bieżącego projektu



3. Tworzenie pustego pliku do wpisania kodu źródłowego programu





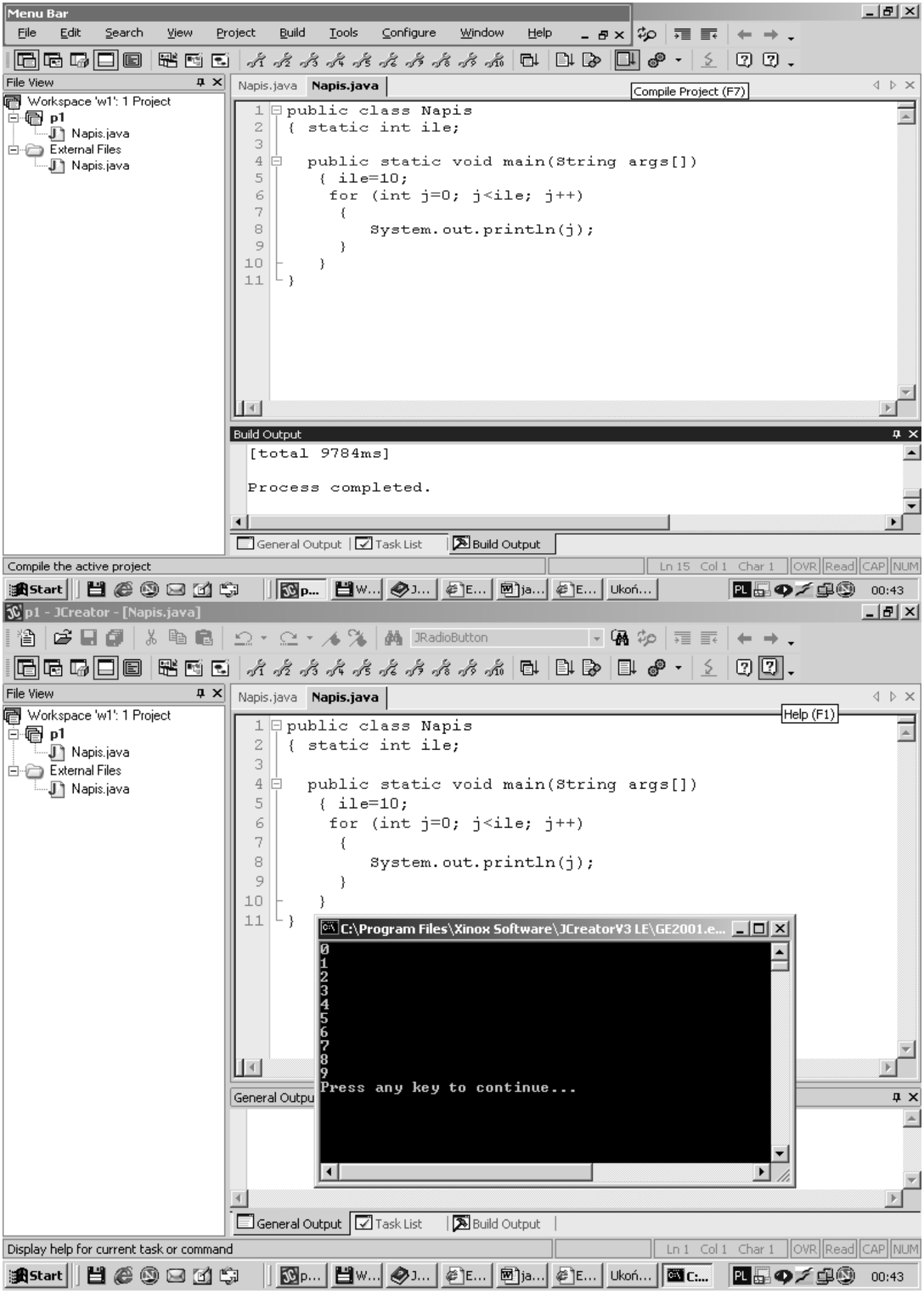


Przykłady

1) Drukowanie na ekranie w pętli wartości typu całkowitego bez tworzenia obiektu

```
public class Napis                    //klasa publiczna niefinalna, nie abstrakcyjna
{
  static int ile;//składowa klasowa (istnieje niezależnie od istnienia obiektu tej klasy)

 public static void main(String args[])
  {
   ile=10;                            //ile musi być składową typu static!
  for (int j=0; j<ile; j++)           //definicja zmiennej sterującej w bloku instrukcji for
   {
     System.out.println(j);           //konwersja zmiennej typu całkowitego na łańcuch
   }                                  //i przejście do następnej linii
  }
}
```


Menu Bar
File Edit Search View Project Build Tools Configure Window Help
File View
Workspace 'w1': 1 Project
p1
Napis.java
External Files
Napis.java
Napis.java Napis.java
Compile Project (F7)
public class Napis
{ static int ile;
public static void main(String args[])
{ ile=10;
for (int j=0; j<ile; j++)
{
System.out.println(j);
}
}
}
Build Output
[total 9784ms]
Process completed.
General Output Task List Build Output
Compile the active project
Ln 15 Col 1 Char 1 OVR Read CAP NUM
Start
00:43
p1 - JCreator - [Napis.java]
JRadioButton
Help (F1)
C:\Program Files\Xinox Software\JCreatorV3 LE\GE2001.e...
0
1
2
3
4
5
6
7
8
9
Press any key to continue...
Display help for current task or command
Ln 1 Col 1 Char 1 OVR Read CAP NUM

2) Drukowanie na ekranie w pętli wartości typu całkowitego z tworzeniem obiektu

```
public class Napis_
{
  int ile;                                  //zmienna składowa klasy

  public static void main(String args[])
   { Napis_ p = new Napis_();               //wywołanie domyślnego konstruktora podczas
                                            //przydziału pamięci na obiekt klasy Napis_
     p.ile=10;                              //p jest referencją do obiektu klasy Napis_
     for (int j=0; j<p.ile; j++)            //odwołanie do obiektu p
      {System.out.println("petla "+j);}     //dodawanie łańcucha pętla do łańcucha
                                            //znaków (cyfry) uzyskanego za pomocą
                                            //konwersji z wartości typu całkowitego j
   }
}
```

3) Wywołanie programu z listą parametrów

java argi Jan Kowalski

```
public class argi                    // klasa publiczna, nie abstrakcyjna i niefinalna
{
  static int ile;                    //składowa klasowa

  public static void main(String args[])
   {
     ile=args.length;                //pobranie liczby parametrów (w przykładzie 2)
                                     // ile musi być składową typu static !
     for (int j=0; j<ile; j++)       //args[0] – Jan (łańcuch bez białych znaków)
                                     //args[1] - Kowalski
     { System.out.println(args[j]);}
  }
}
```

---

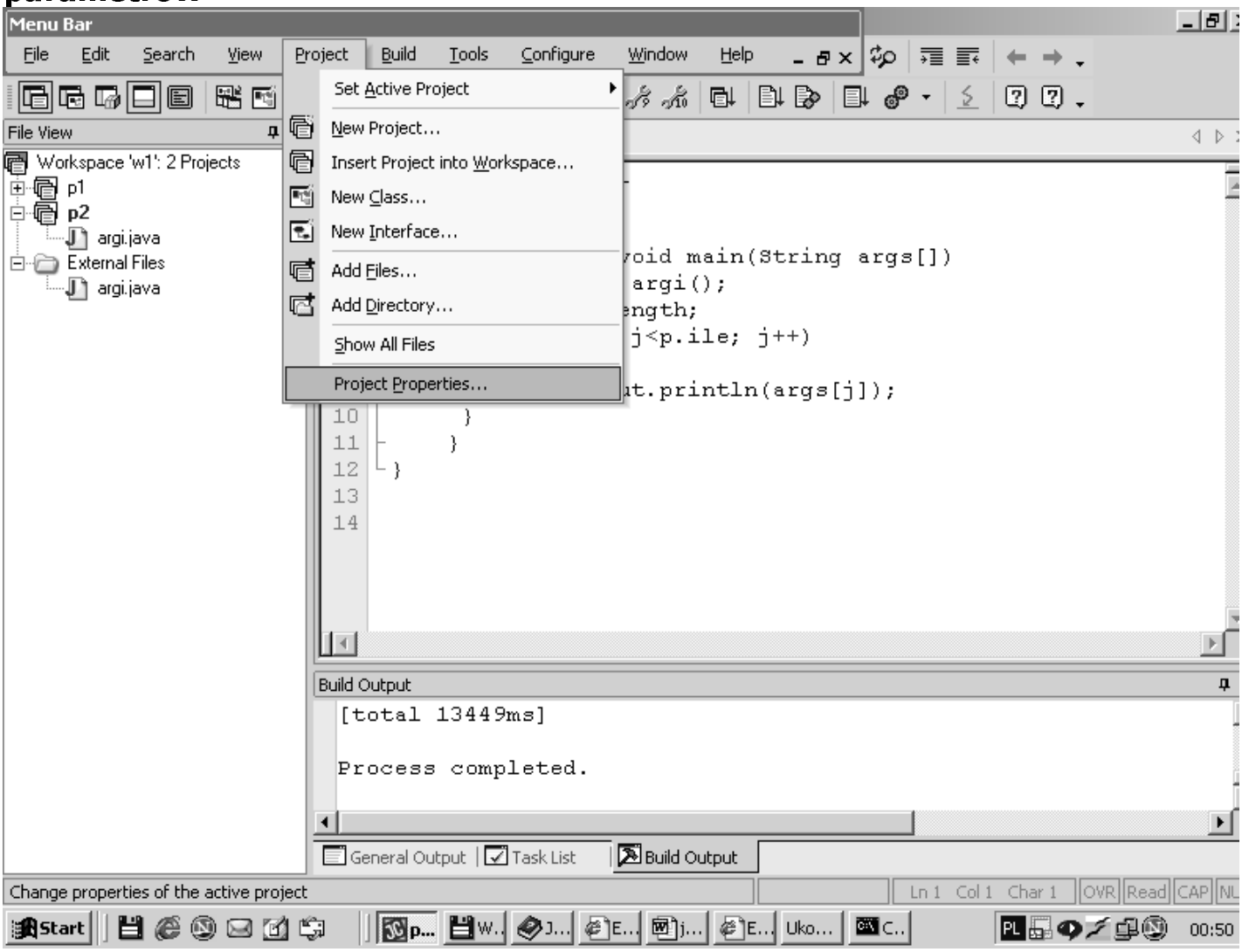
4) Wywołanie programu z listą parametrów

java argi_ Jan Kowalski

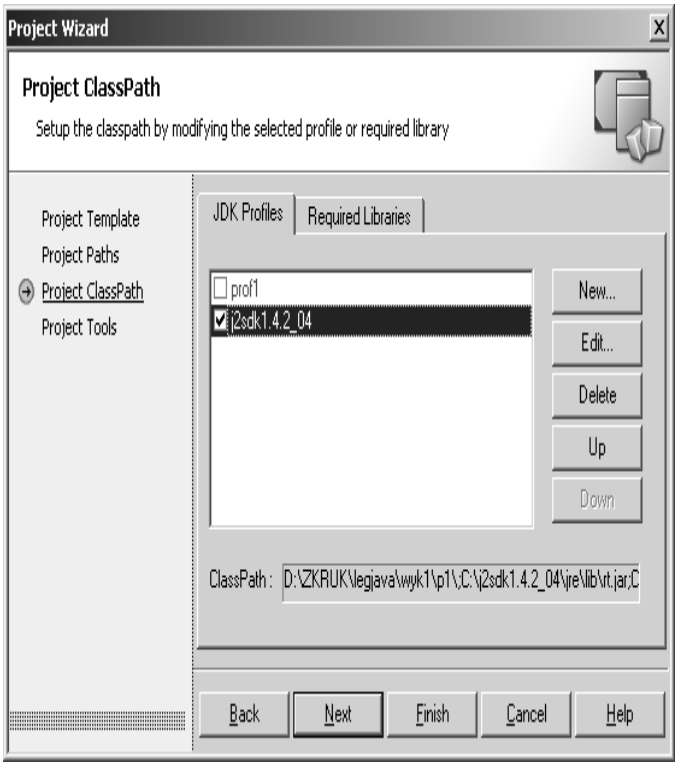
```
public class argi_
{
   int ile;

   public static void main(String args[])
   {
    argi_ p = new argi_();
    p.ile=args.length;
    for (int j=0; j<p.ile; j++)
    { System.out.println(args[j]);}
  }
}
```

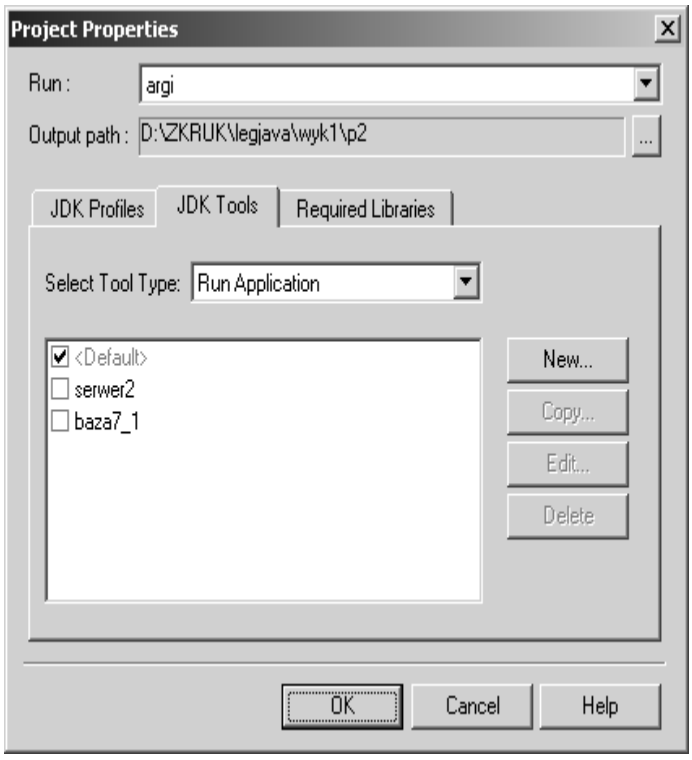
Ustawienie środowiska JCreator do uruchamiania programu z listą parametrów

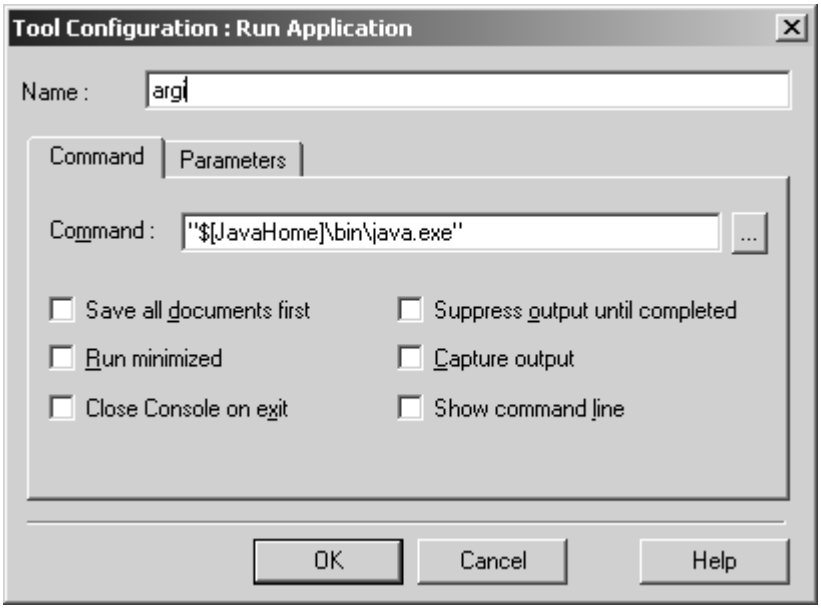


Zgłoszenie się domyślnej zakładki JDK Profiles

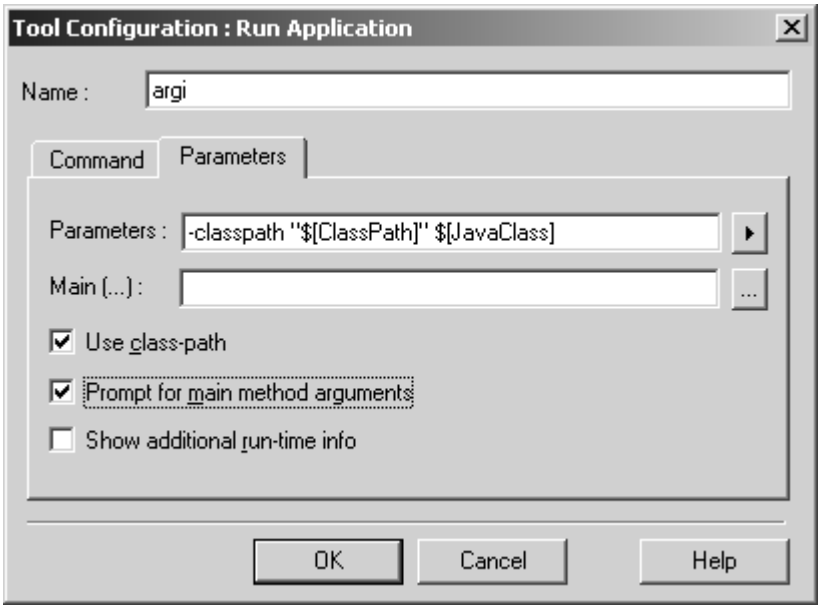


Po wyborze zakładki JDK Tools

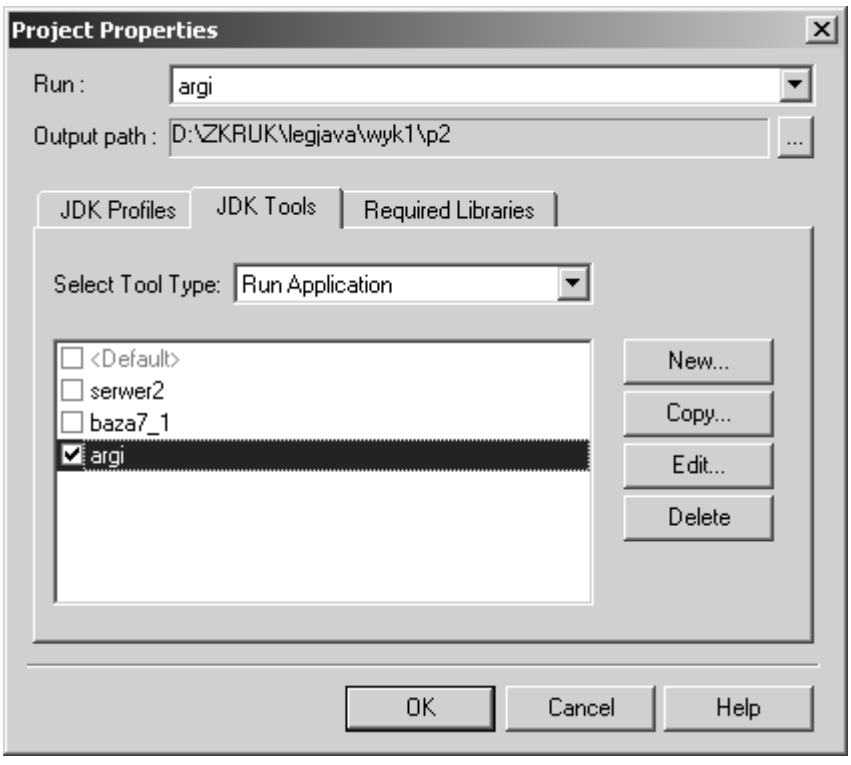




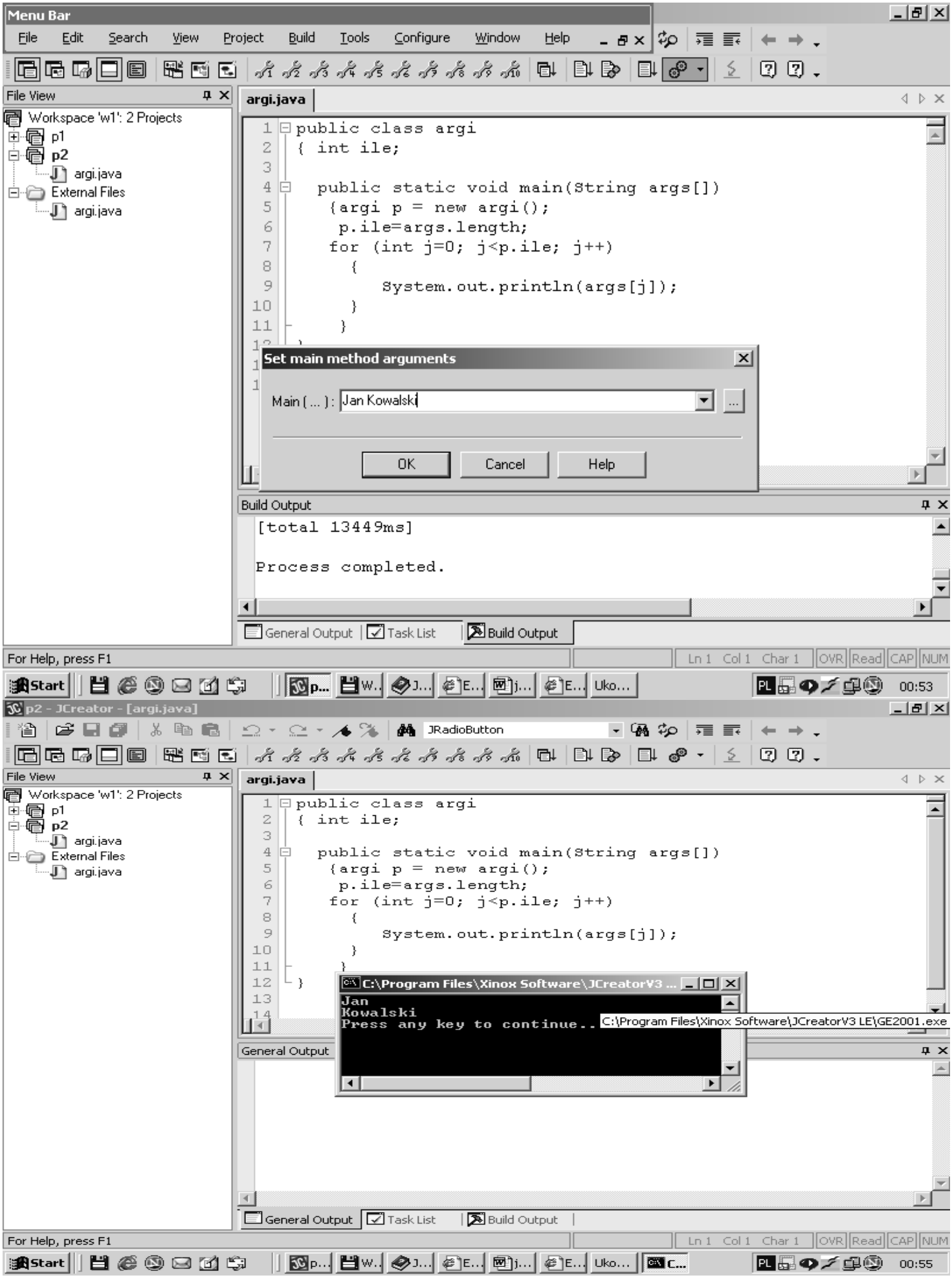
Przejście do domyślnej zakładki *Command* i nazwanie pliku konfiguracyjnego *argi*



Przejście do domyślnej zakładki *Parameters* i wybór opcji *Prompt for main metho arguments*



Przypisanie pliku konfiguracyjnego *argi* do bieżącego projektu


Menu Bar
File Edit Search View Project Build Tools Configure Window Help
File View
Workspace 'w1': 2 Projects
p1
p2
argi.java
External Files
argi.java
argi.java
public class argi
{ int ile;
public static void main(String args[])
{argi p = new argi();
p.ile=args.length;
for (int j=0; j<p.ile; j++)
{
System.out.println(args[j]);
}
}
Set main method arguments
Main (...) :
Jan Kowalski
OK
Cancel
Help
Build Output
[total 13449ms]
Process completed.
General Output
Task List
Build Output
For Help, press F1
Ln 1 Col 1 Char 1 OVR Read CAP NUM
Start
00:53
p2 - JCreator - [argi.java]
JRadioButton
C:\Program Files\Xinox Software\JCreatorV3 ...
Jan
Kowalski
Press any key to continue..
C:\Program Files\Xinox Software\JCreatorV3 LE\GE2001.exe
00:55