

Laboratorinis darbas Nr.2: Programavimo įrankiai – modulinis testavimas naudojant JUnit.

Darbo užduotis:

Naudojant antro laboratorinio darbo metų įgytas žinias, realizuoti modulinio testavimo atvejį dėstytojo pateiktos sistemos nurodytam metodui.

Reikalavimai:

- Modulinis testas turi būti realizuotas [Java](#) kalba naudojant [JUnit](#) testavimo karkasą.
- Testas turi būti realizuotas ir paleidžiamas [Eclipse](#) įrankio aplinkoje.
- Testas turi apimti visus galimus testuojamo metodo vykdymo scenarijus bei įėjimo duomenų aibę – t.y. sugadinus sistemos testuojamą metodą testas turi parodyti pranešimą apie klaidą.
- Moduliniai testai turi būti laikomi atskirai nuo programos kodo pakete „tests“.

Užduoties atlikimo scenarijus

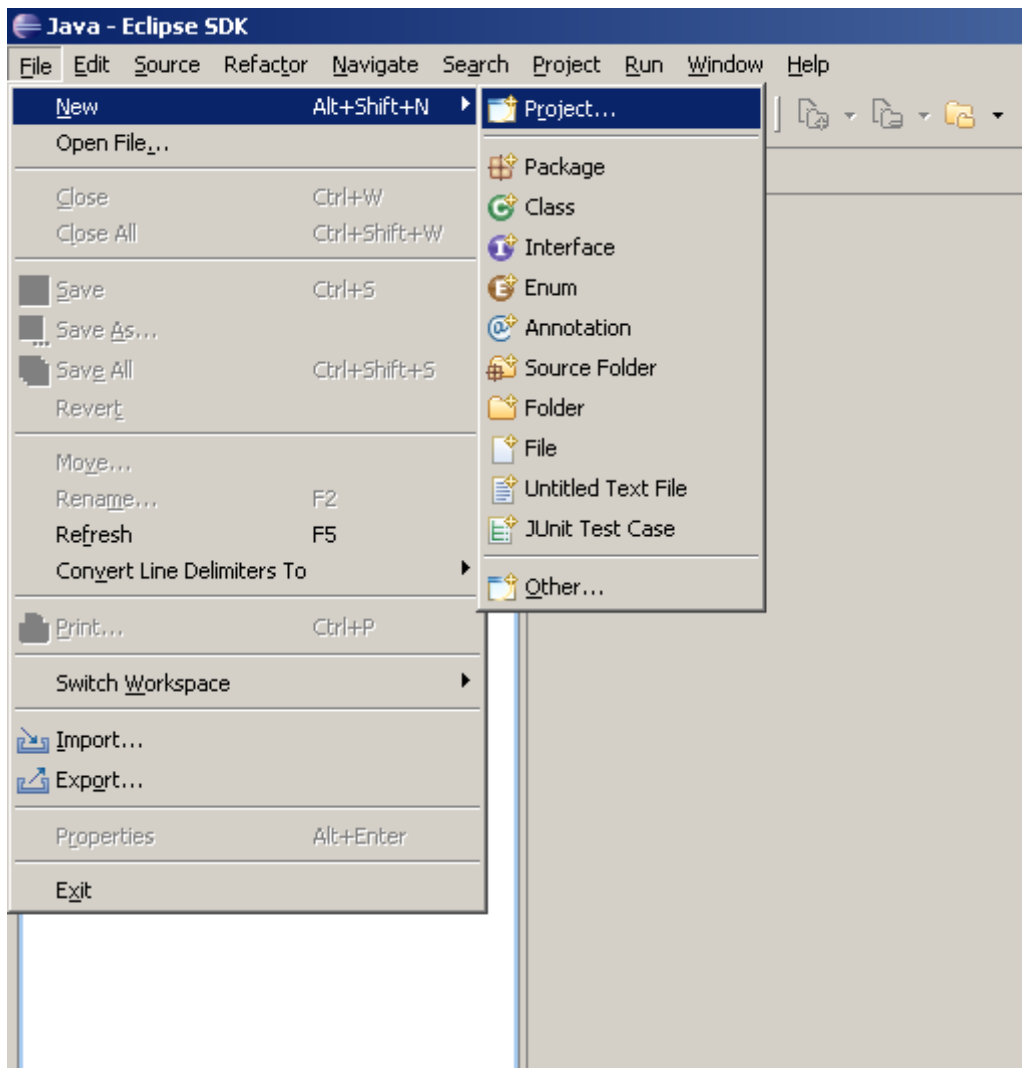
1. Sukurti naują Java projektą Eclipse programų kūrimo aplinkoje. Projekto pavadinimas - Jūsų pažymėjimo numeris.
2. Importuoti į sukurtą projektą dėstytojo nurodytos programos išeities failus.
3. Sukompiliuoti importuotos programos išeities failus.
4. Importuoti į projektą modulinio testavimo JUnit biblioteką (JUnit.jar).
5. Sukurti ir realizuoti modulinį testą dėstytojo nurodytam programos metodui.
6. Pademonstruoti sukurtą modulinį testą dėstytojui.

Užduoties atlikimo pavyzdys sistemai „Shapes“ (Figūros)

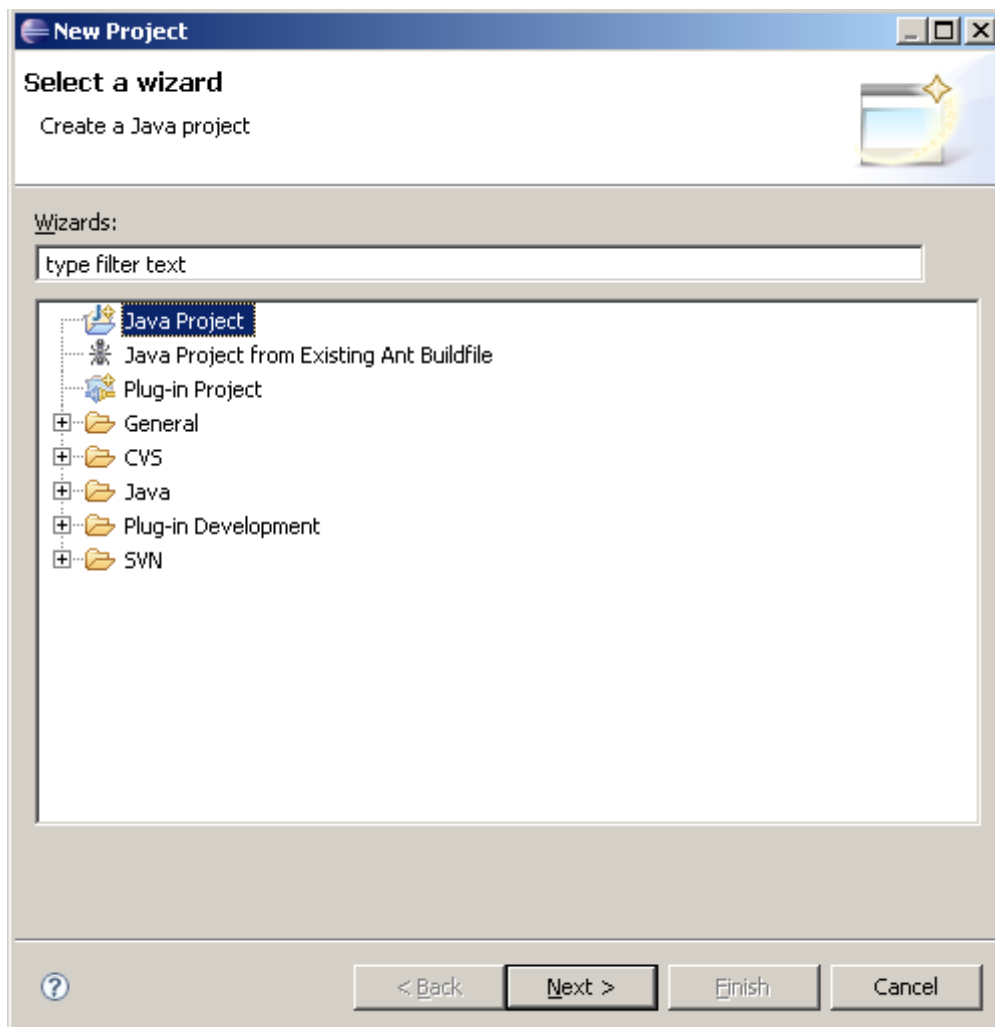
Naujo projekto sukūrimas Eclipse aplinkoje

Eclipse yra nemokama laisvai prieinama programų kūrimo aplinka. Ją galima parsisiųsti iš www.eclipse.org tinklalapio.

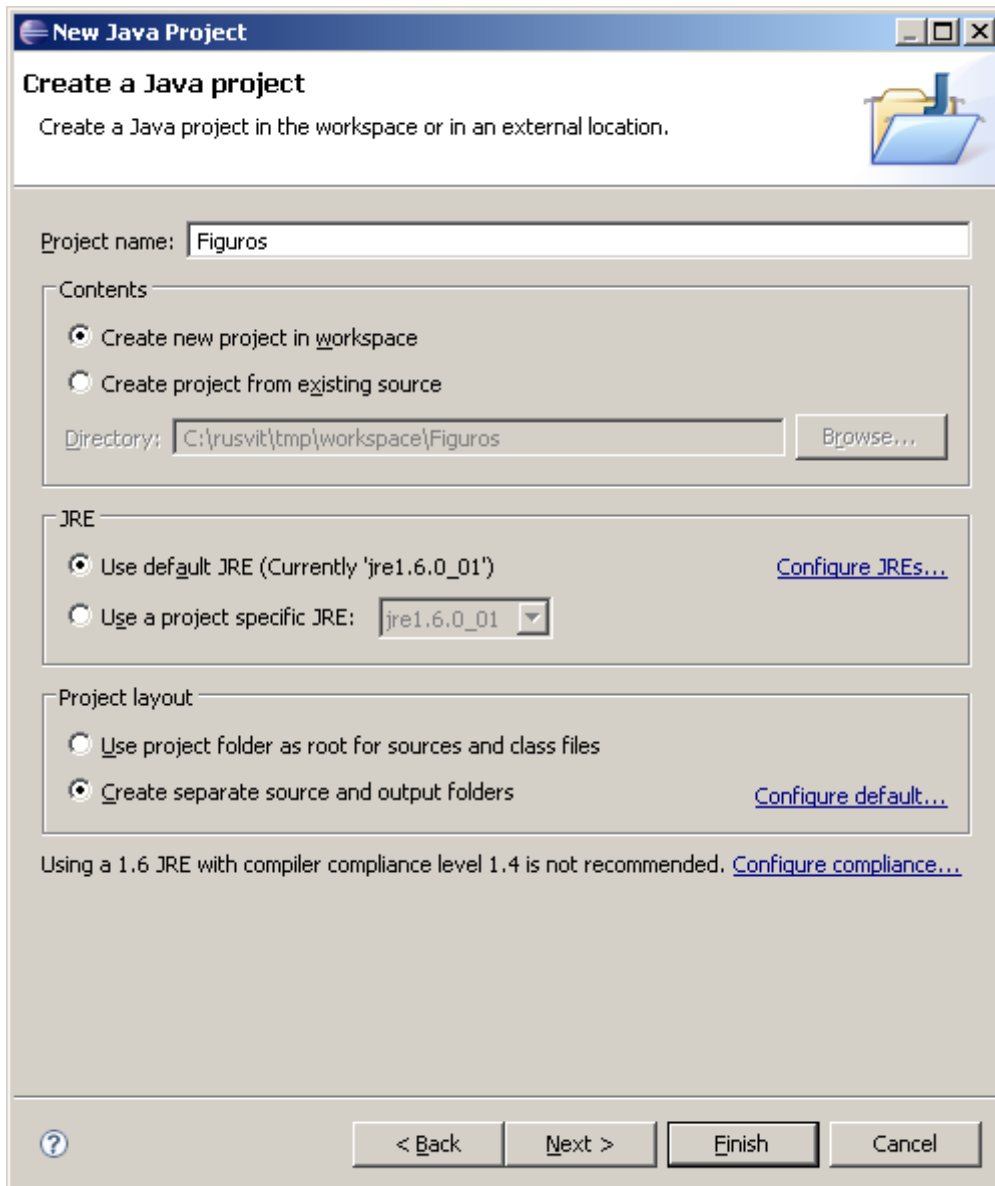
Paleiskite Eclipse įrankį. Sukurkite naują projektą kaip parodyta Pav. 1. Projekto tipo pasirinkimo lange (Pav. 2) pasirinkite *Java Project* ir spauskite mygtuką *Next*. *New Java Project* lange (Pav. 3) į laukelį *Project name* įrašykite projekto pavadinimą *Figuros*, pažymėkite *Create separate source and output folders* punktą jei jis dar nepažymėtas ir spauskite *Finish* mygtuką. Sukurtą projektą galima matyti *Package Explorer* kortelėje. Jei šios kortelės nematote atverkite ją naudojant pagrindinį meniu *Window* → *Show View* → *Package Explorer*.



Pav. 1: Naujo projekto sukūrimas Eclipse aplinkoje



Pav. 2: Projekto tipo pasirinkimo langas



Pav. 3: Naujo Java projekto kūrimo vedlio langas

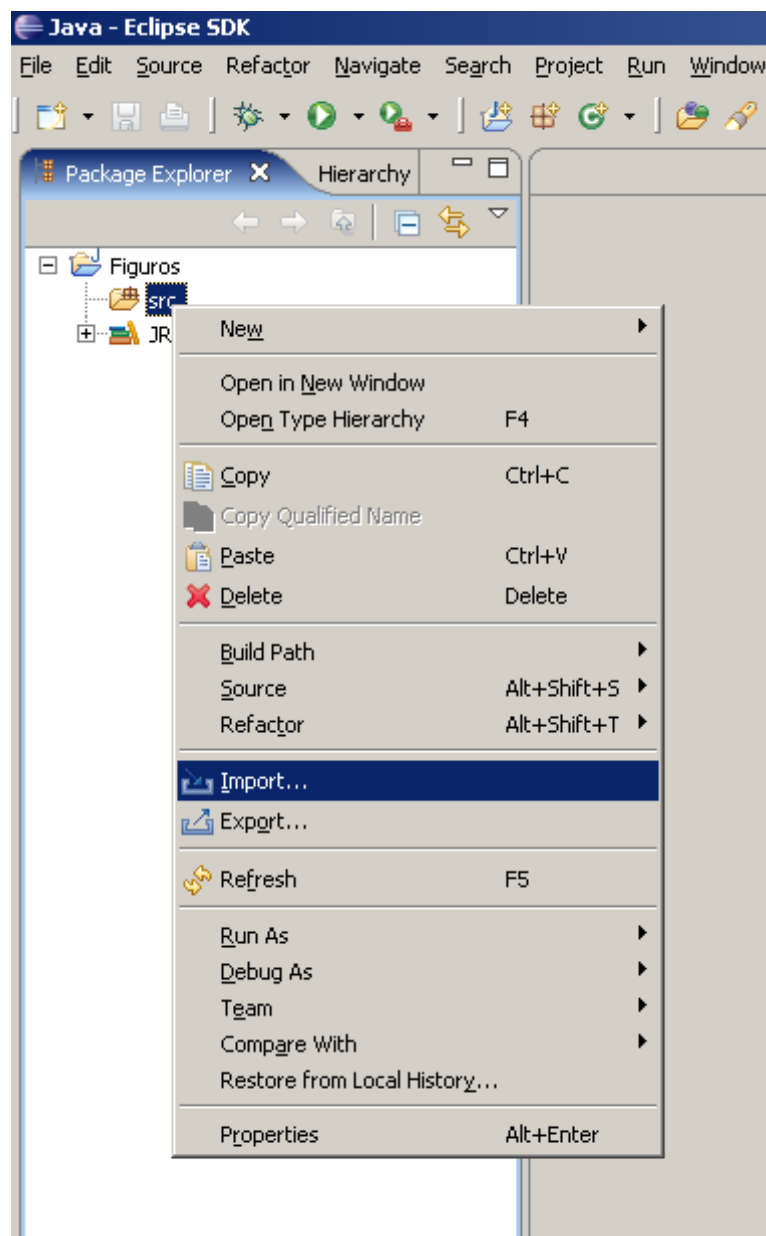
Testuojamos programos išėities failų importavimas į sukurta projektą

Parsisiųskite testuojamos programos Shapes suarchyvuotus išėities failus naudodamiesi šia nuoroda:

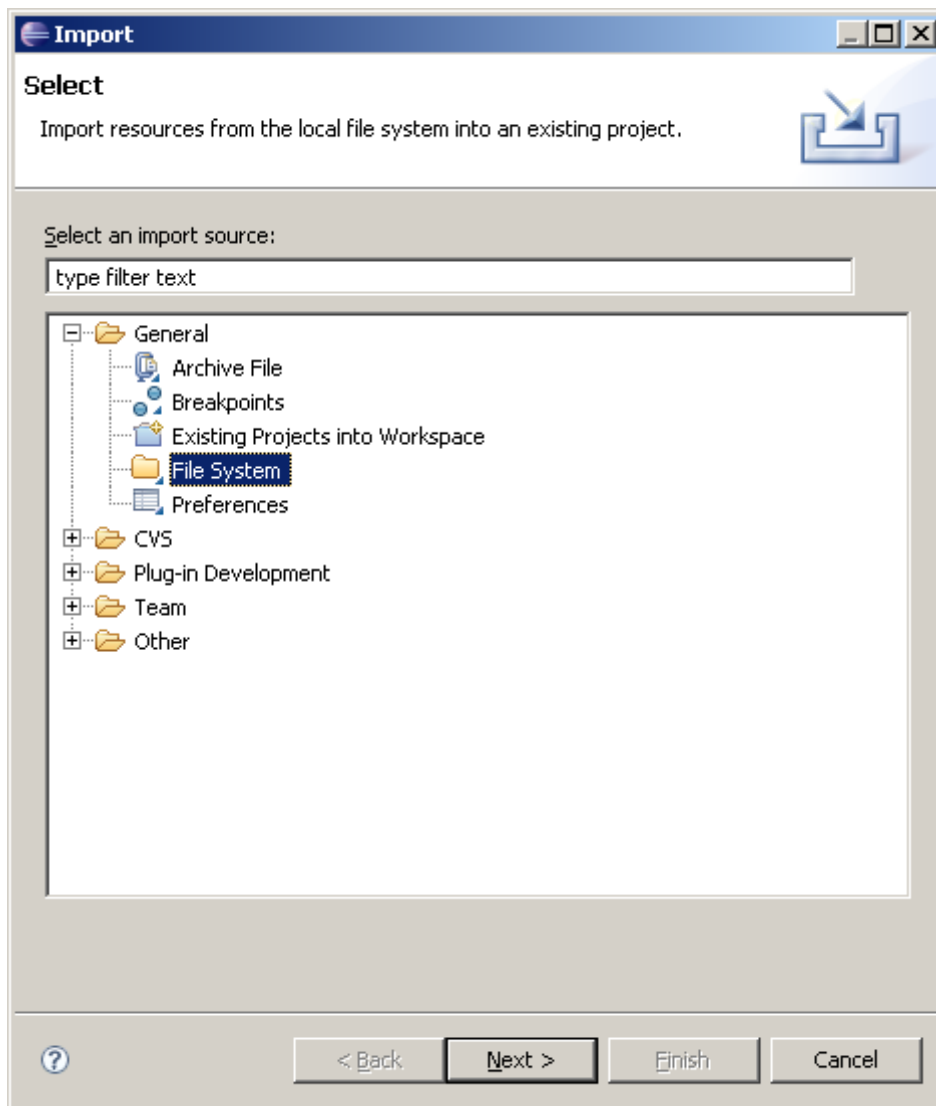
http://vaidila.vdu.lt/~i6ruvi/pikm/files/lab2_shapes.zip

Išarchyvuokite parsiaustą archyvą (atsiminkite failų buvimo vietą).

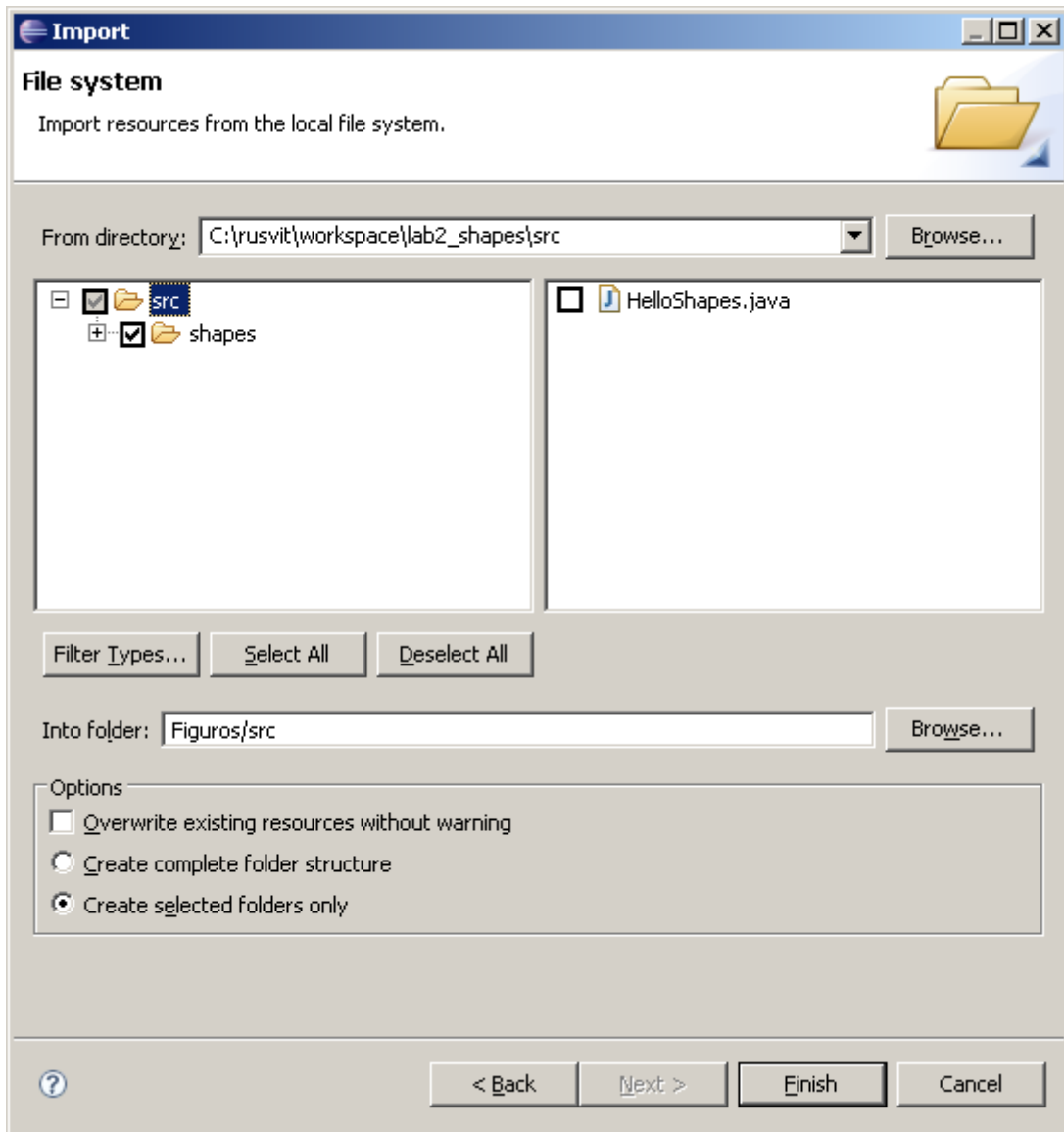
Dešiniu pelės klavišu spragtelėkite ant projekto viduje esančio *src* katalogo ir iš kontekstinio menu pasirinkite *Import* (Pav. 4). Importuojamų duomenų tipo pasirinkimo lange (Pav. 5) pasirinkite *File System*. Failų sistemos importavimo lange (Pav. 6) *From directory* laukelyje nurodykite kelią iki išarchyvuotų programos failų *src* katalogo. Pažymėkite *shapes* katalogą importavimui (importuojamo *src* katalogo viduje). Pažymėkite *Create selected folders only* punktą jei jis nepažymėtas ir spauskite *Finish* mygtuką. Jūsų projekto *src* katalogo viduje bus sukurtas *shapes* paketas su importuotais programos kodu (Pav. 7). Jeigu Jūsų projekto struktūra neatitiks struktūros parodytos paveiksle 7 kompiliatorius negalės sukompiliuoti programos išėities failų. Atsiras kompiliavimo klaida dėl failų viduje esančios *package* signatūros aprašymo ir tikros failo buvimo vietos nesutapimo. Ši problema gali būti sprendžiama pataisant *package* signatūrą pagal failo buvimo vietą arba failas perkeliamas į *package* signatūros aprašytą katalogą.



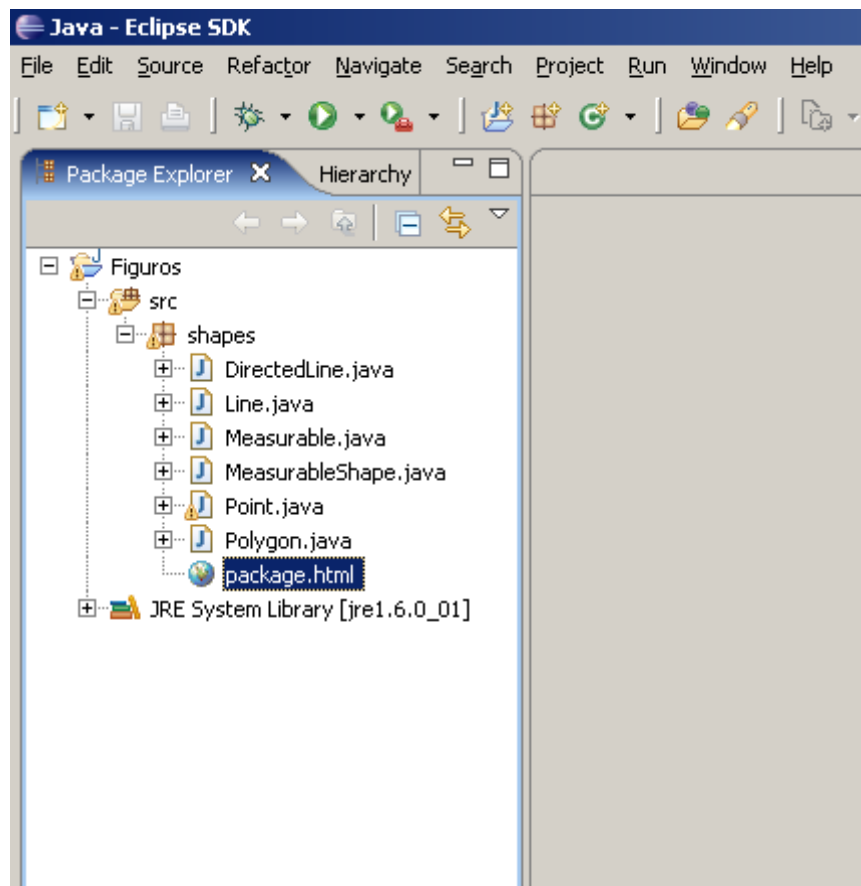
Pav. 4: Java programos kodo importavimas



Pav. 5: Importuojamų duomenų tipo pasirinkimo langas



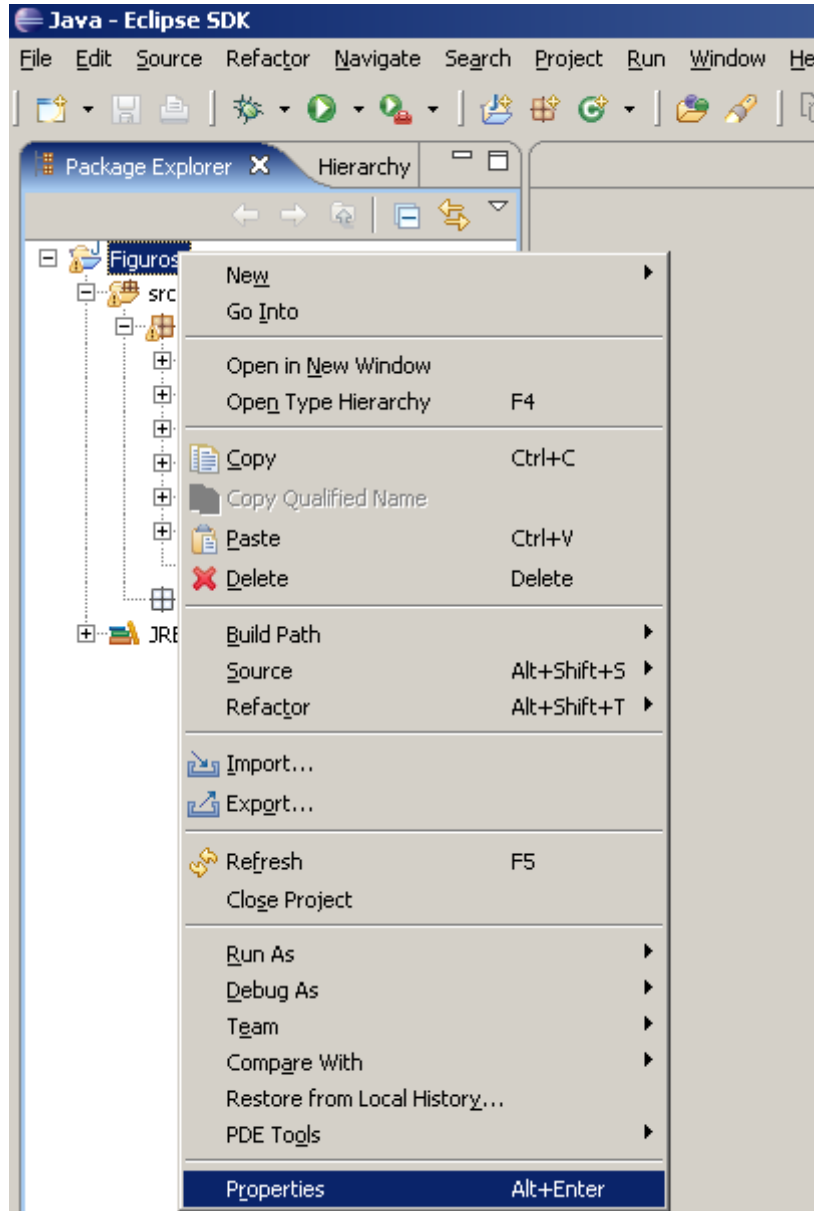
Pav. 6: Failų sistemos importavimo langas



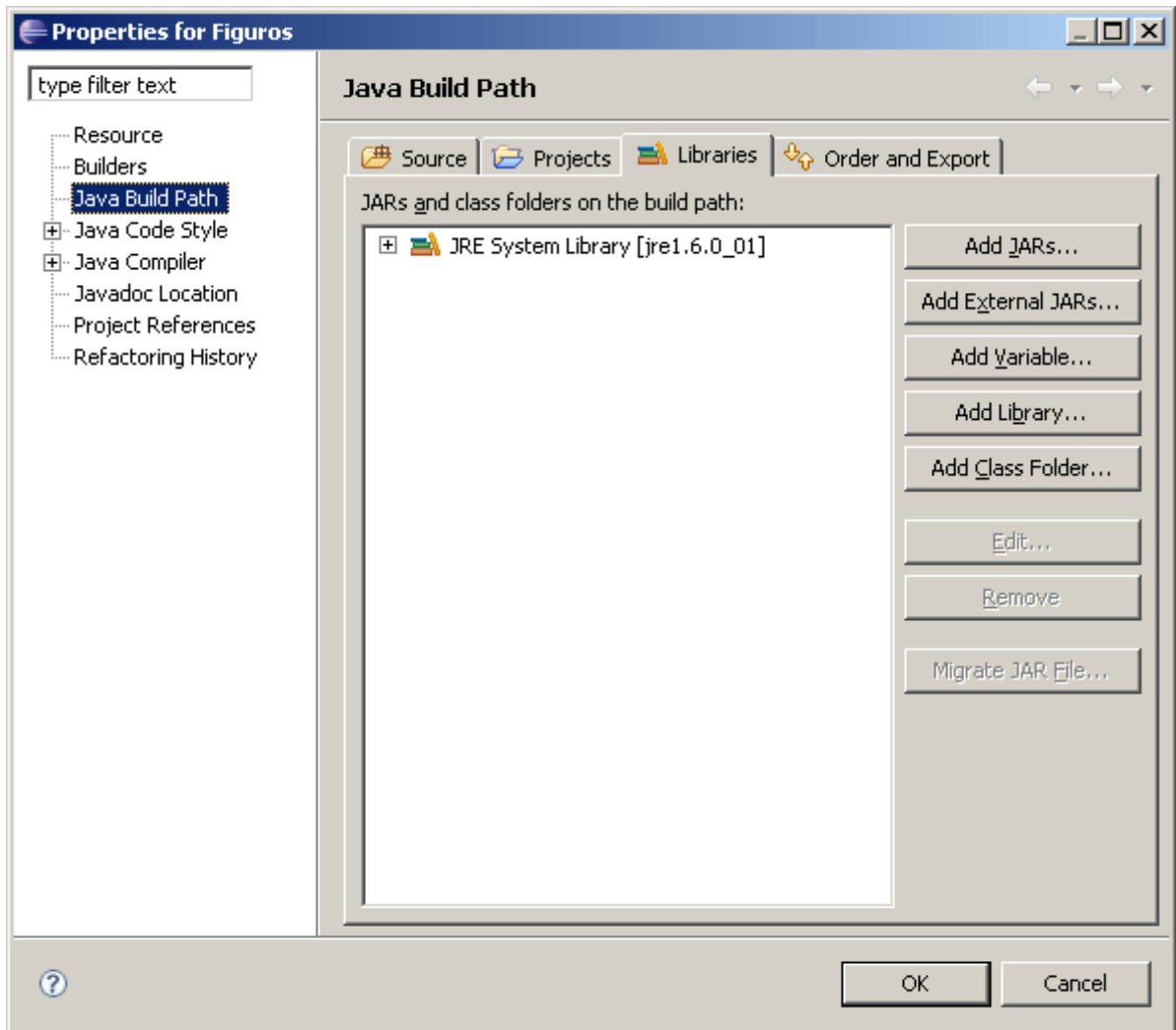
Pav. 7: *shapes* paketas sėkmingai importuotas į projektą

JUnit Testavimo karkaso bibliotekos importavimas į projektą.

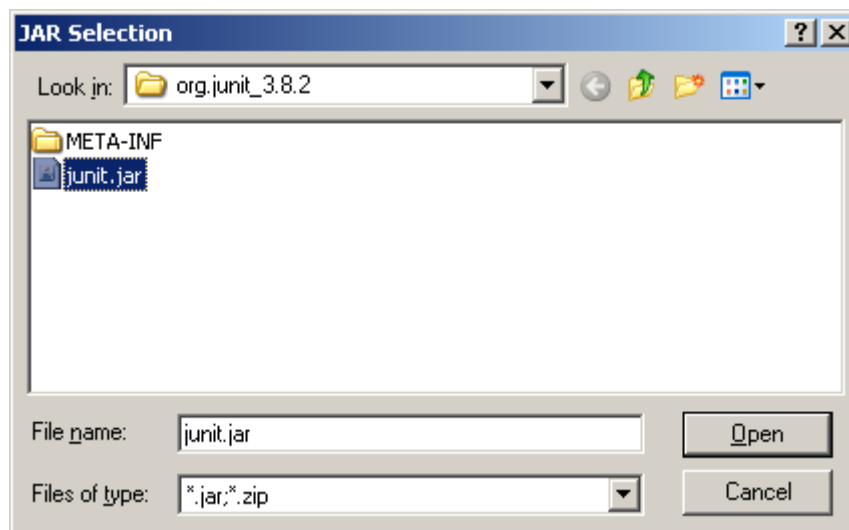
Norėdami kurti JUnit modulinius testus reikia į projektą įtraukti JUnit biblioteką. Projekto savybių langas iššaukiamas projekto kontekstinio meniu (Pav. 8) pagalba. Projekto savybių lango kairėje pasirenkam *Java Build Path* bei atverčiam *Libraries* kortelę (Pav. 9). Norėdami įtrauktų papildomą JUnit biblioteką paspaudžiam *Add External JARs* mygtuką. Bibliotekos pasirinkimo lango (Pav. 10) pagalba surandam ir įtraukiam *junit.jar* biblioteką į projekto papildomų bibliotekų sąrašą (Pav. 11). *Junit.jar* biblioteką galima rasti Eclipse įrankio *plugins* kataloge arba parsisiūsti iš <http://www.junit.org>.



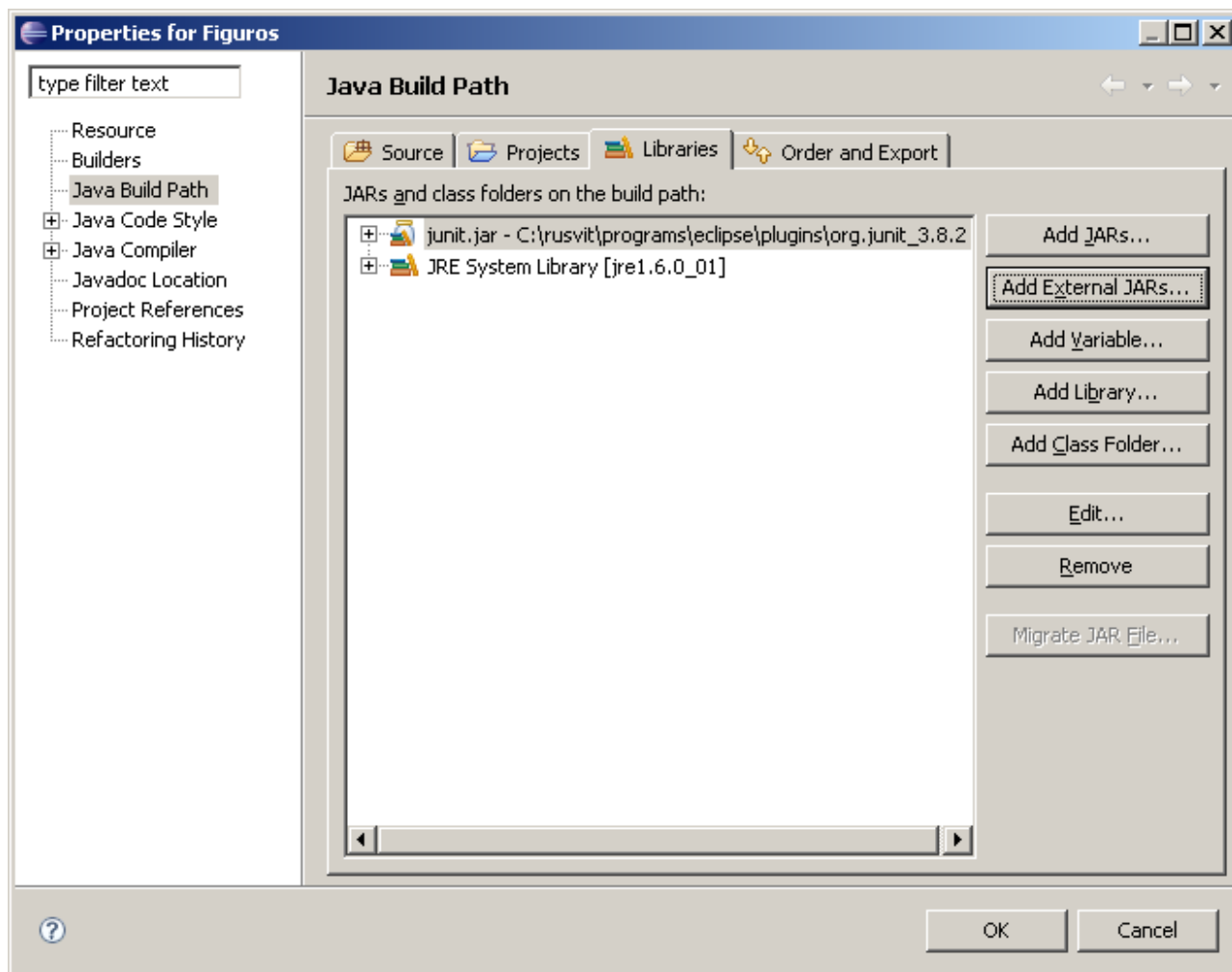
Pav. 8: Projekto savybių lango iššaukimas



Pav. 9: Projekto savybių langas



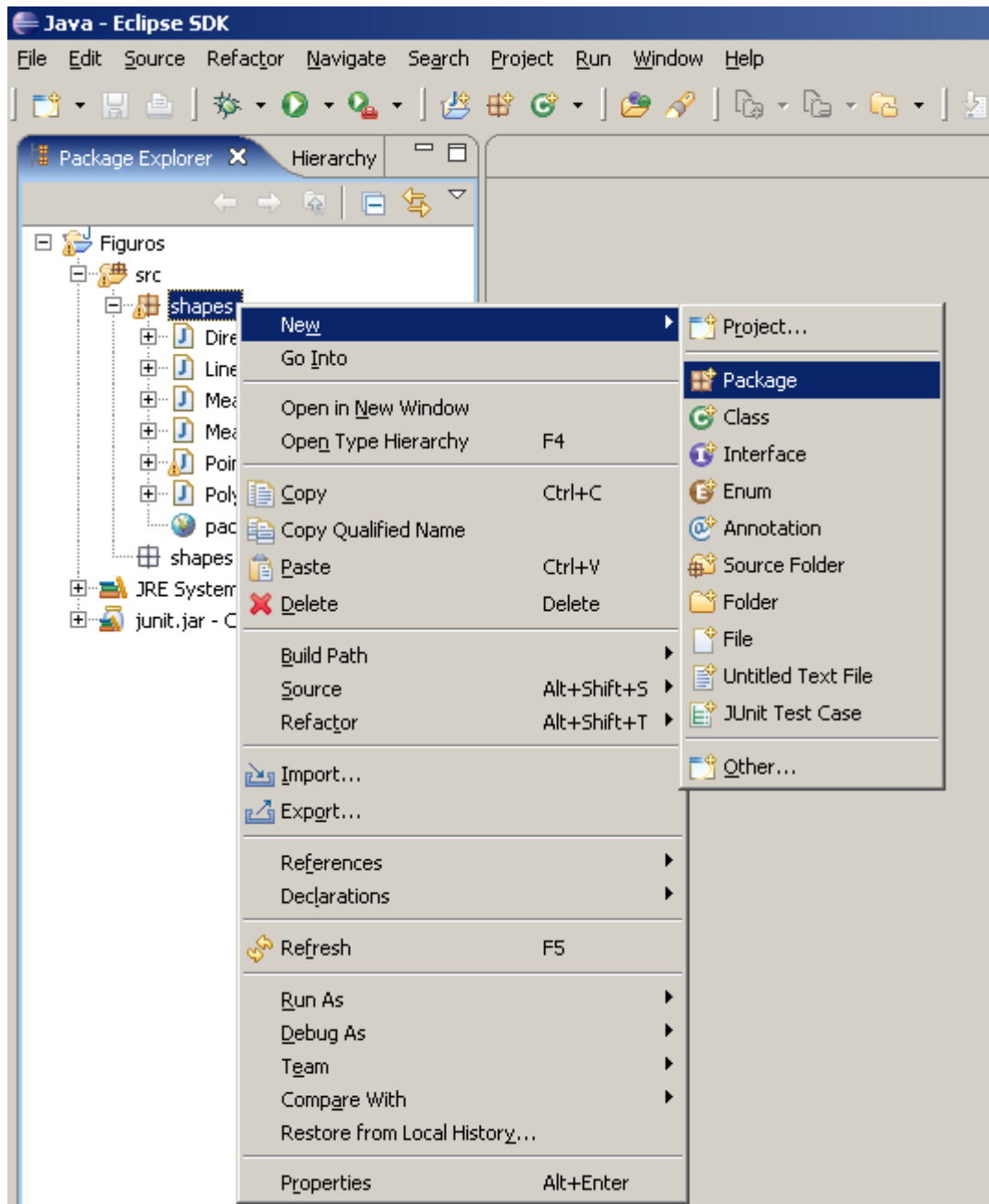
Pav. 10: Papildomos bibliotekos pasirinkimo langas



Pav. 11: Projekto papildomų bibliotekų sąrašas

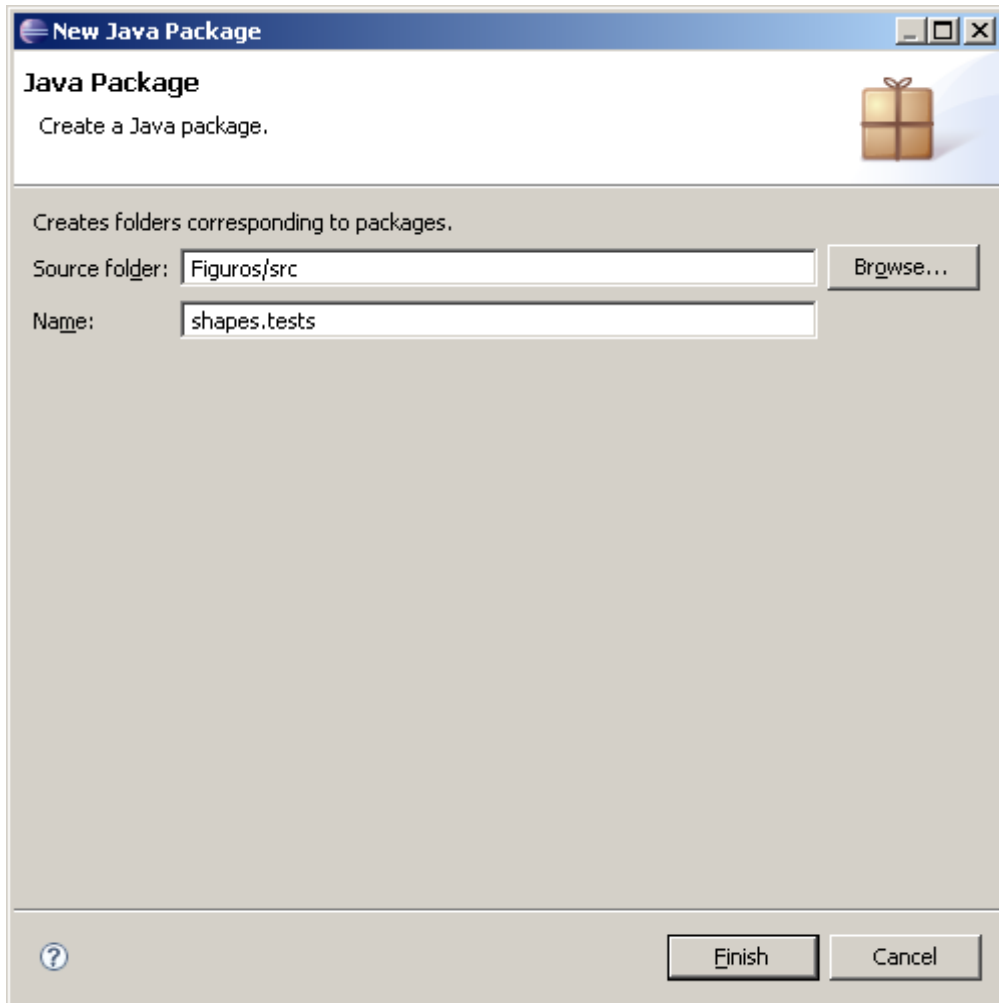
JUnit Modulinio testo realizavimas klasės *Point equals* metodu

Sukurkite naują paketą *tests shapes* paketo viduje. Paketo *tests* kūrimui naudokite *shapes* paketo kontekstinį meniu (Pav. 12).



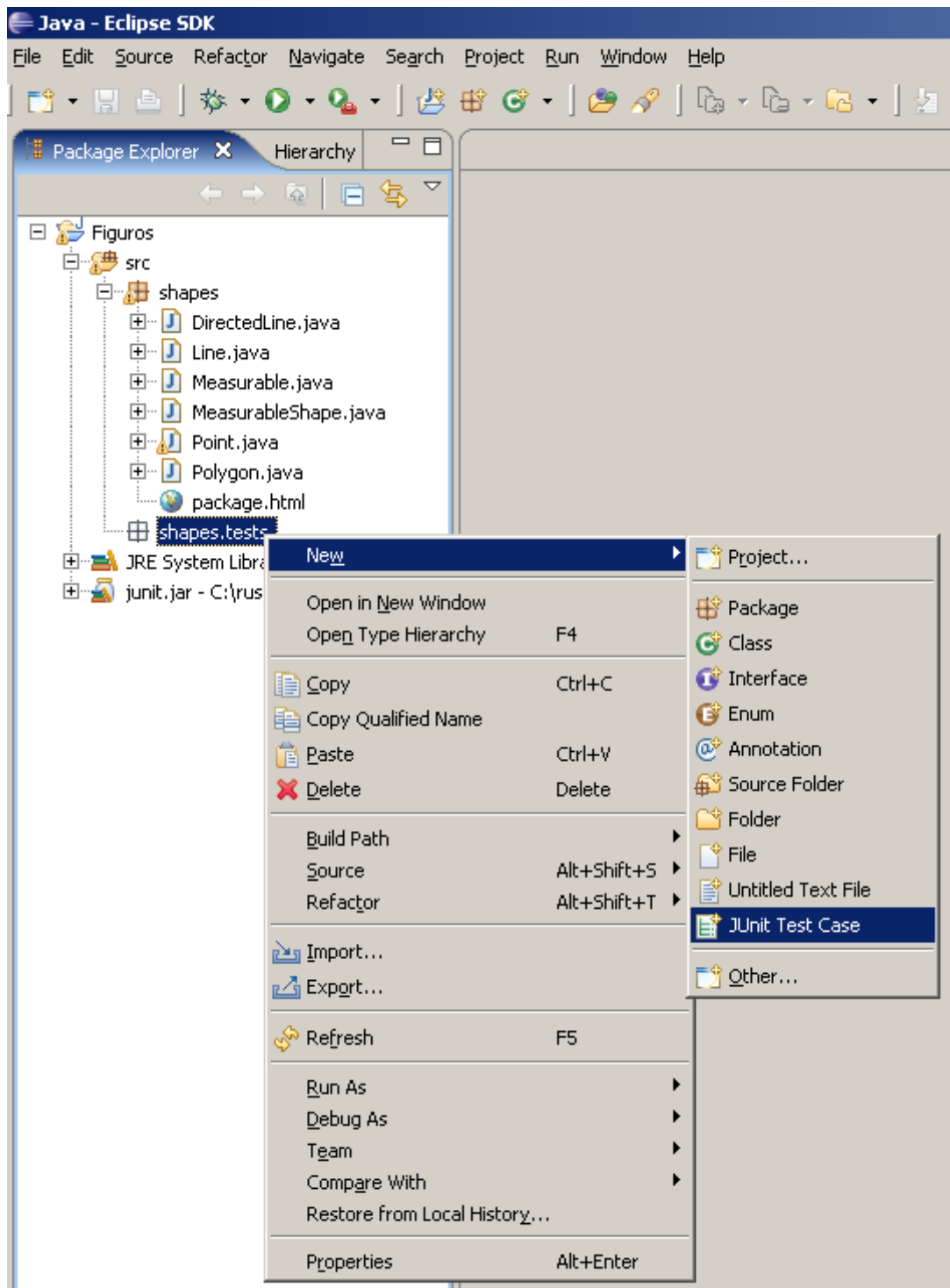
Pav. 12: Naujo paketo sukūrimas

Naujo paketo kūrimo lange laukelyje *Name* įveskite paketo pilną pavadinimą *shapes.tests* (Pav. 13).



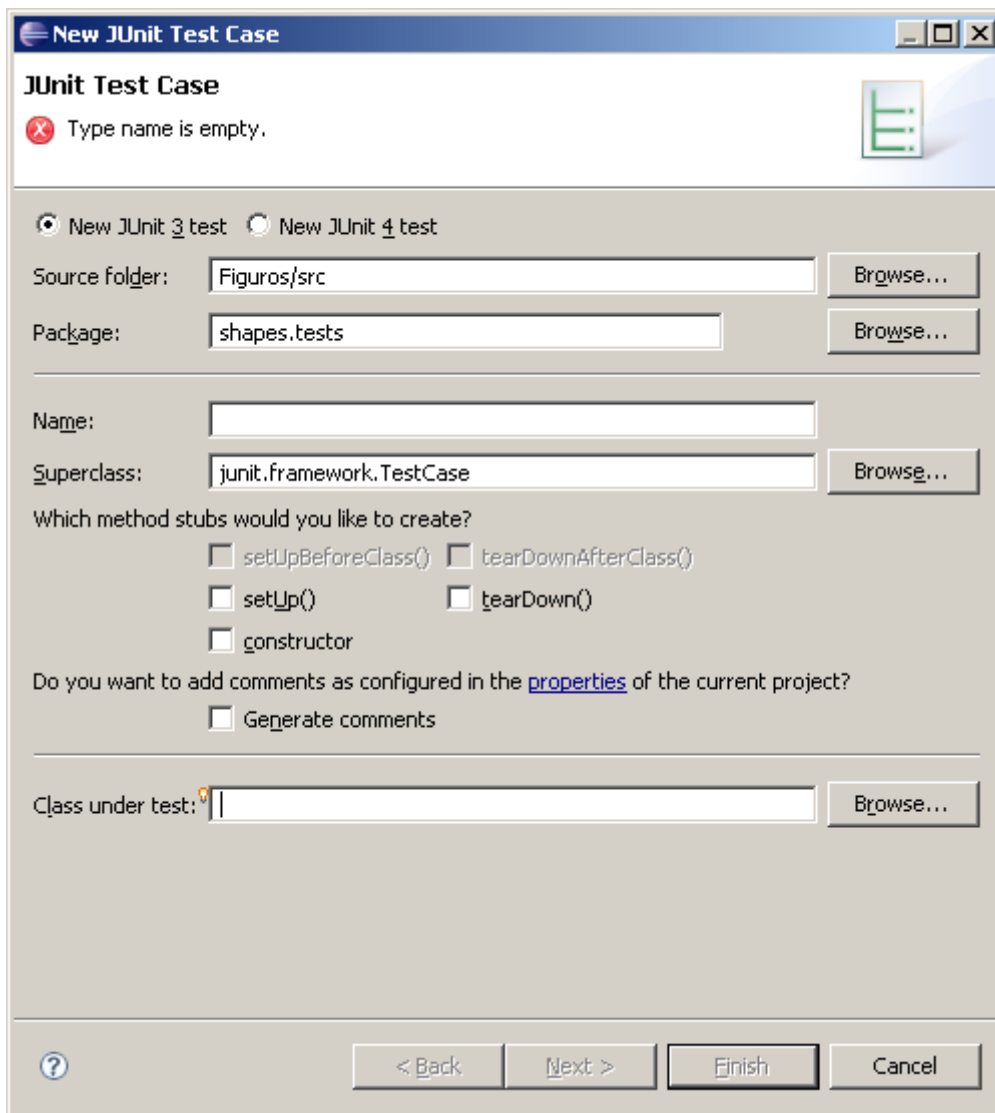
Pav. 13: Naujo paketo kūrimo langas

Sukurto paketo viduje kontekstinio menu pagalba sukurkite naują JUnit testavimo atvejį (JUnit Test Case Pav. 14).

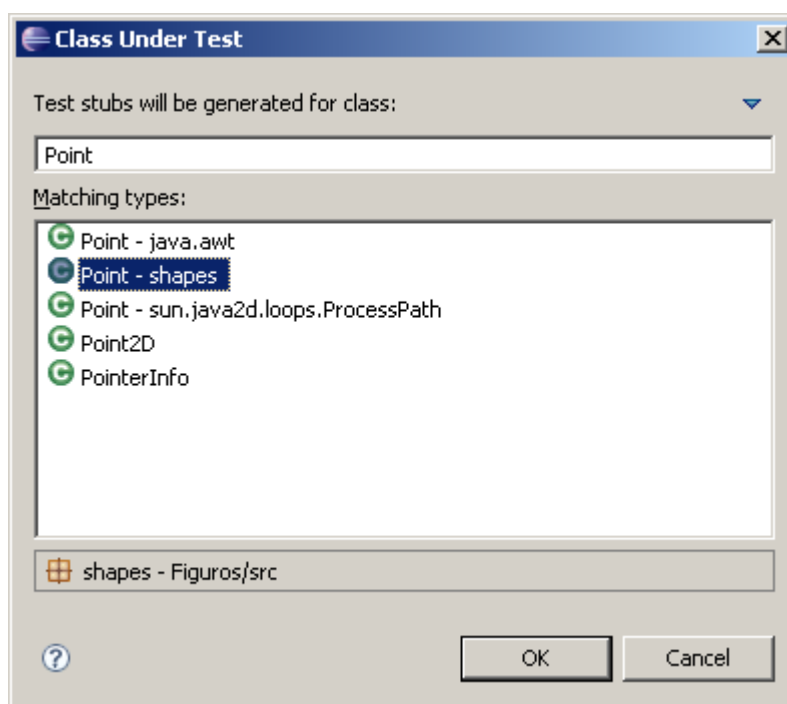


Pav. 14: Naujo testavimo atvejo sukūrimas

Naujo testavimo atvejo kūrimo lange (Pav. 15) į laukelį *Name* įveskite testavimo klasės pavadinimą *PointTest*. Ties *Class under test* lauko paspauskite mygtuką *Browse* ir klasės pasirinkimo lange (Pav. 15) įveskite *Point* (pasirinkite testuojamą klasę *Point* iš *shapes* katalogo).



Pav. 15: Naujo testavimo atvejo kūrimo langas



Pav. 16: Testuojamos klasės pasirinkimo langas

Pasirinkę testuojamą klasę spauskite *Next* mygtuką.

New JUnit Test Case

Select the name of the new JUnit test case. You have the options to specify the class under test and on the next page, to select methods to be tested.

New JUnit 3 test New JUnit 4 test

Source folder:

Package:

Name:

Superclass:

Which method stubs would you like to create?

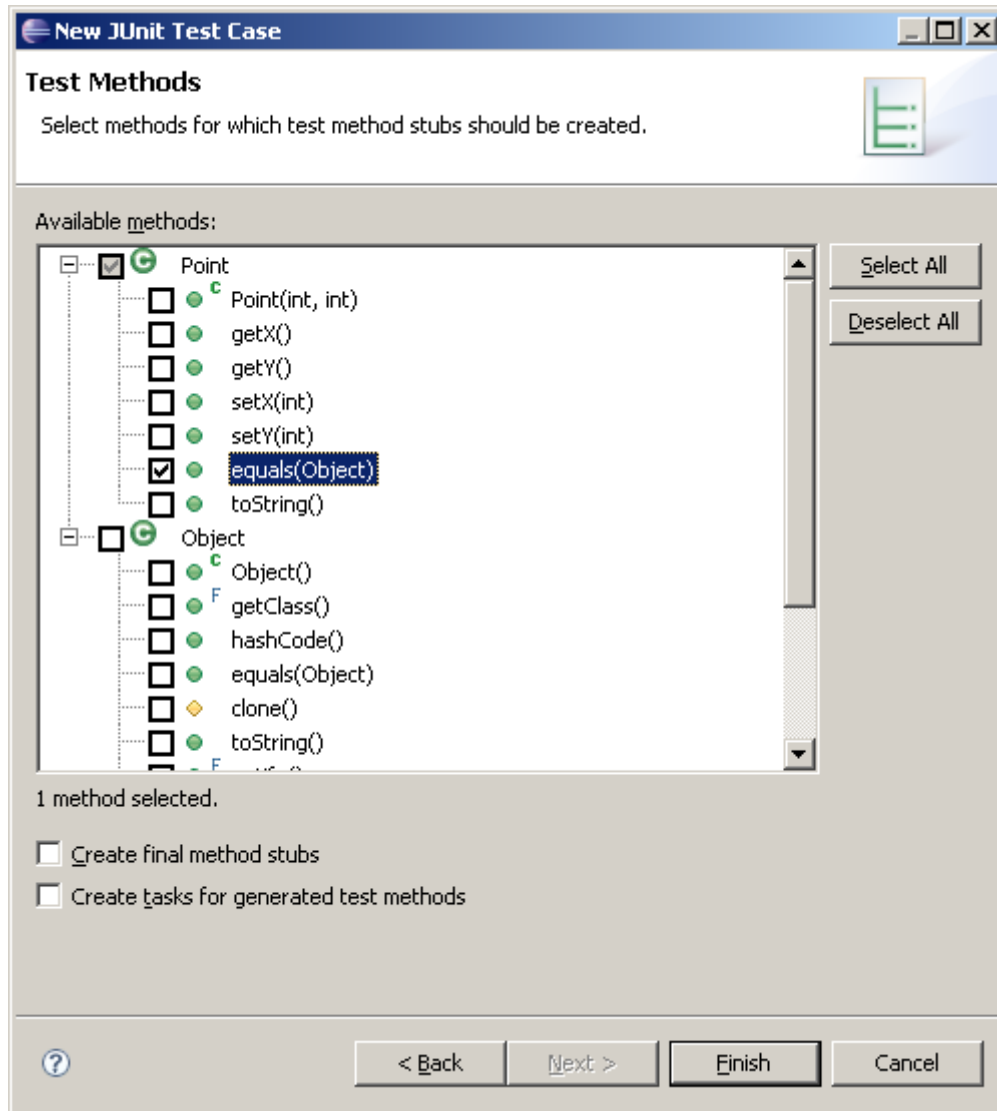
setUpBeforeClass() tearDownAfterClass()
 setUp() tearDown()
 constructor

Do you want to add comments as configured in the [properties](#) of the current project?
 Generate comments

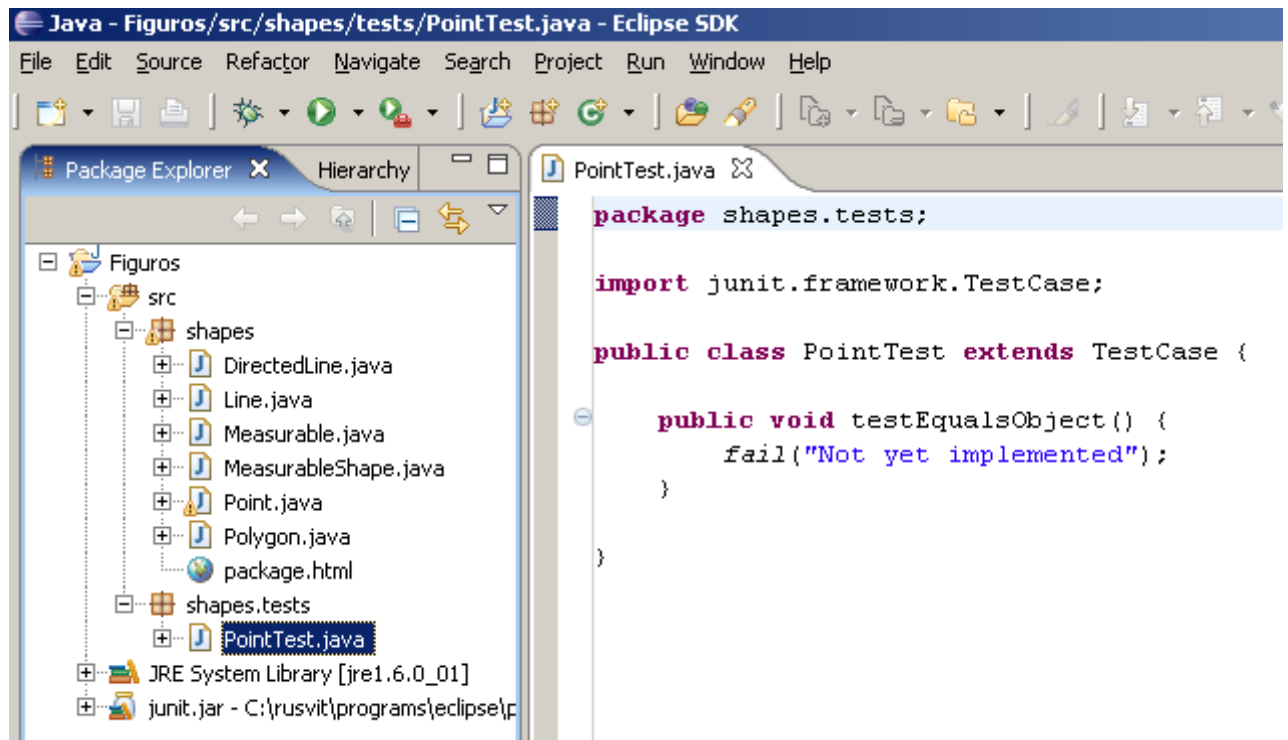
Class under test:

Pav. 17: Testavimo klasės PointTest kūrimas

Testuojamų metodų pasirinkimo lange (Pav. 18) pažymėkite *equals* metodą ir spauskite *Finish*. Eclipse įrankis sukurs modulinio testavimo klasę *PointTest* su metodu *testEqualsObject* skirtu klasės *Point* metodui *equals* testuoti (Pav. 19).



Pav. 18: Testuojamų metodų pasirinkimo langas



Pav. 19: Eclipse įrankio sukurtas pradinis testas

Žemiau pateikta pavyzdinė *testEqualsObject* metodo realizacija. Šią realizaciją galima nukopijuoti į sukurto metodo vidų. Dėl trūkstamos klasės *Point* import įrašo atsiras kompiliavimo. Šią klaidą galima ištaisyti pele spragtelėjus lemputės paveiksluką klaidos kairėje (Pav. 22). Pateiktame galimų klaidos sprendimų sąraše pasirinkus *import Point (shapes)* Eclipse įterps reikiamą import įrašą ir kompiliavimo klaida išnyks.

```

public void testEqualsObject() {

    int x = 3, y = 7;

    //sukuriam du vienuodus taskus
    Point point1 = new Point(x, y);
    Point point2 = new Point(x, y);

    //istestuojam ar metodas equals metodus grazina true kai taskai vienodi
    assertTrue("Taskai vienodi equals turejo grazinti 'true'", point1.equals(point2));

    //nustatom taskui point2 x koordinate didesne per viena
    point2.setX(x+1);

    //dabar taskai point1 ir point2 yra nebevienodi
    assertFalse("Taskai skirtingi, equals turejo grazinti 'false'", point1.equals(point2));

    //atstatom pradine x reiksme
    point2.setX(x);
    //istestuojam ar taskai vel vienodi
    assertTrue("Taskai vienodi equals turejo grazinti 'true'", point1.equals(point2));

    //dabar pakeiciam y koordinate
    point2.setY(y-1);

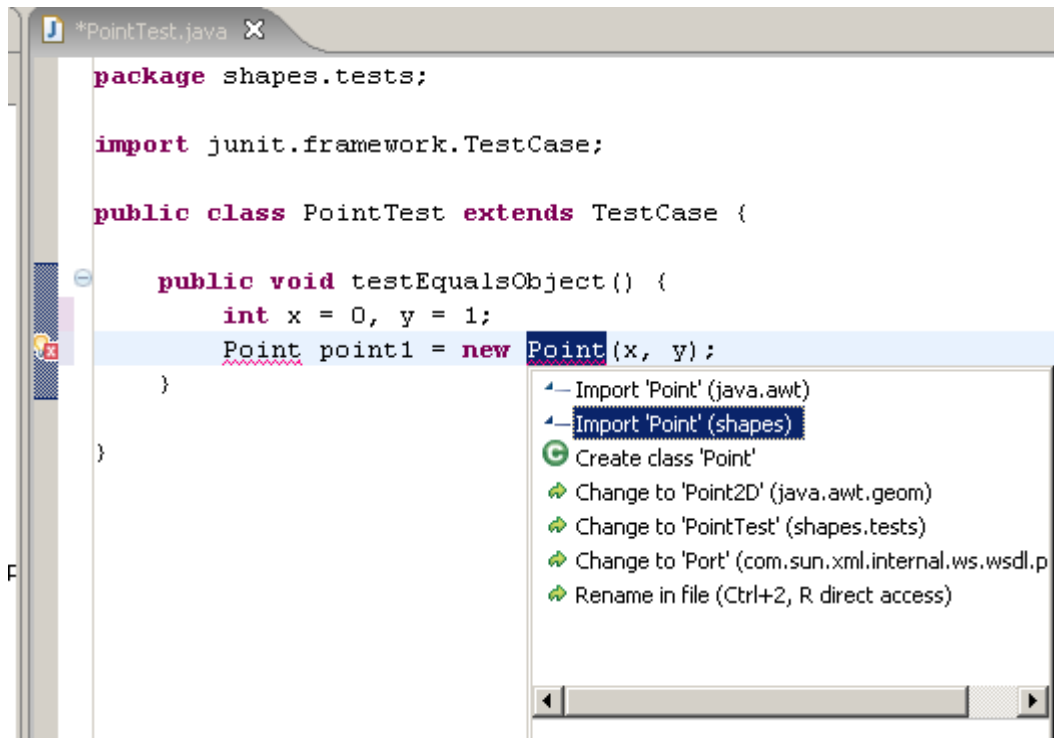
    //dabar taskai point1 ir point2 yra nebevienodi
    assertFalse("Taskai skirtingi, equals turejo grazinti 'false'", point1.equals(point2));

    //patikrinam ar tasku palyginimo metodas equals veikia su null tipo iejimo duomenimis
    assertFalse("equals turejo grazinti 'false' kai lyginama su 'null'", point1.equals(null));

    //patikrinam ar tasku palyginimo metodas equals veikia su kitokio nei Point tipo iejimo duomenimis
    assertFalse("equals turejo grazinti 'false' kai lyginama su 'String'", point1.equals(new String()));
}

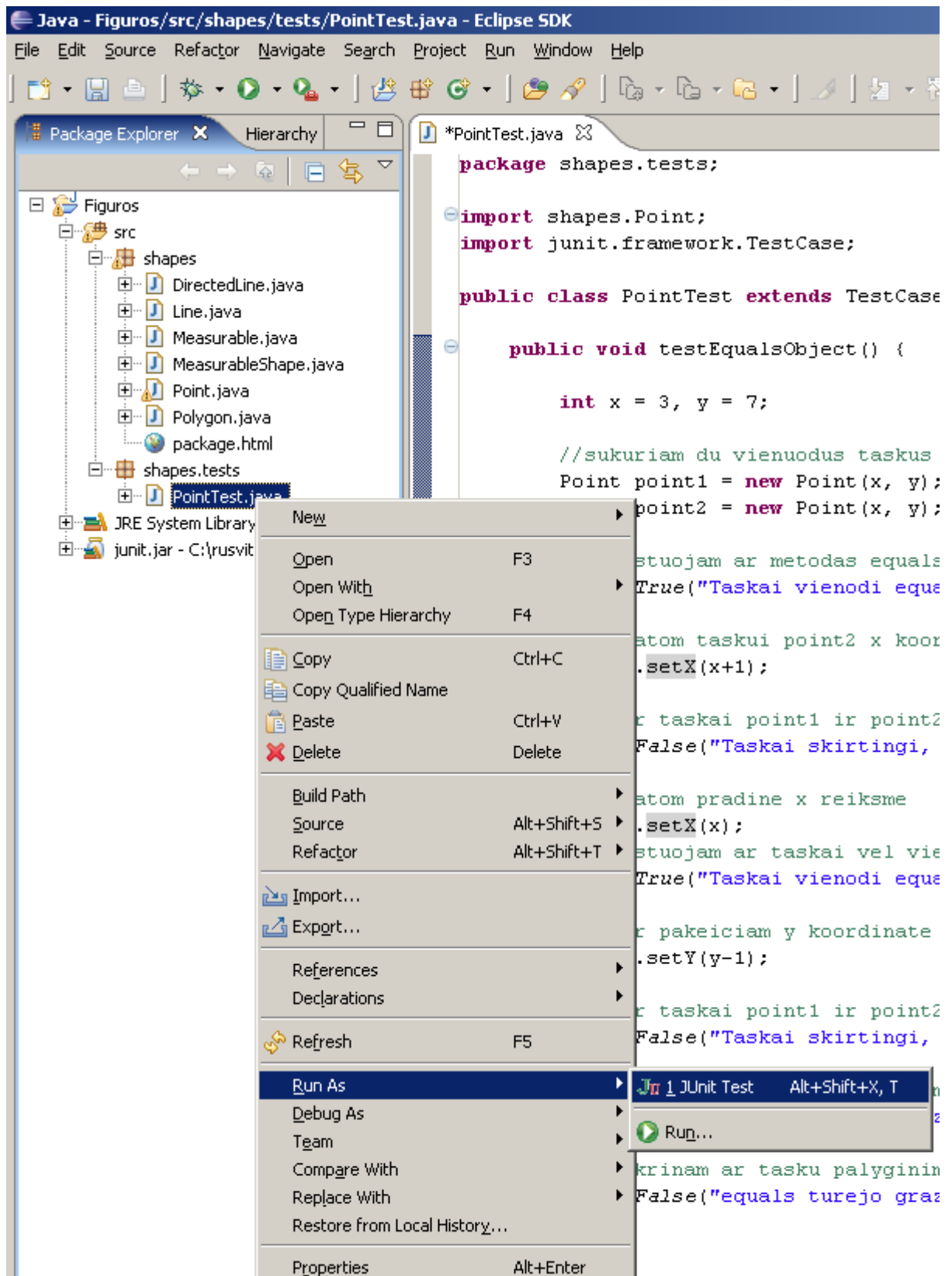
```

Pav. 20: Metodo *testEqualsObject* realizacija

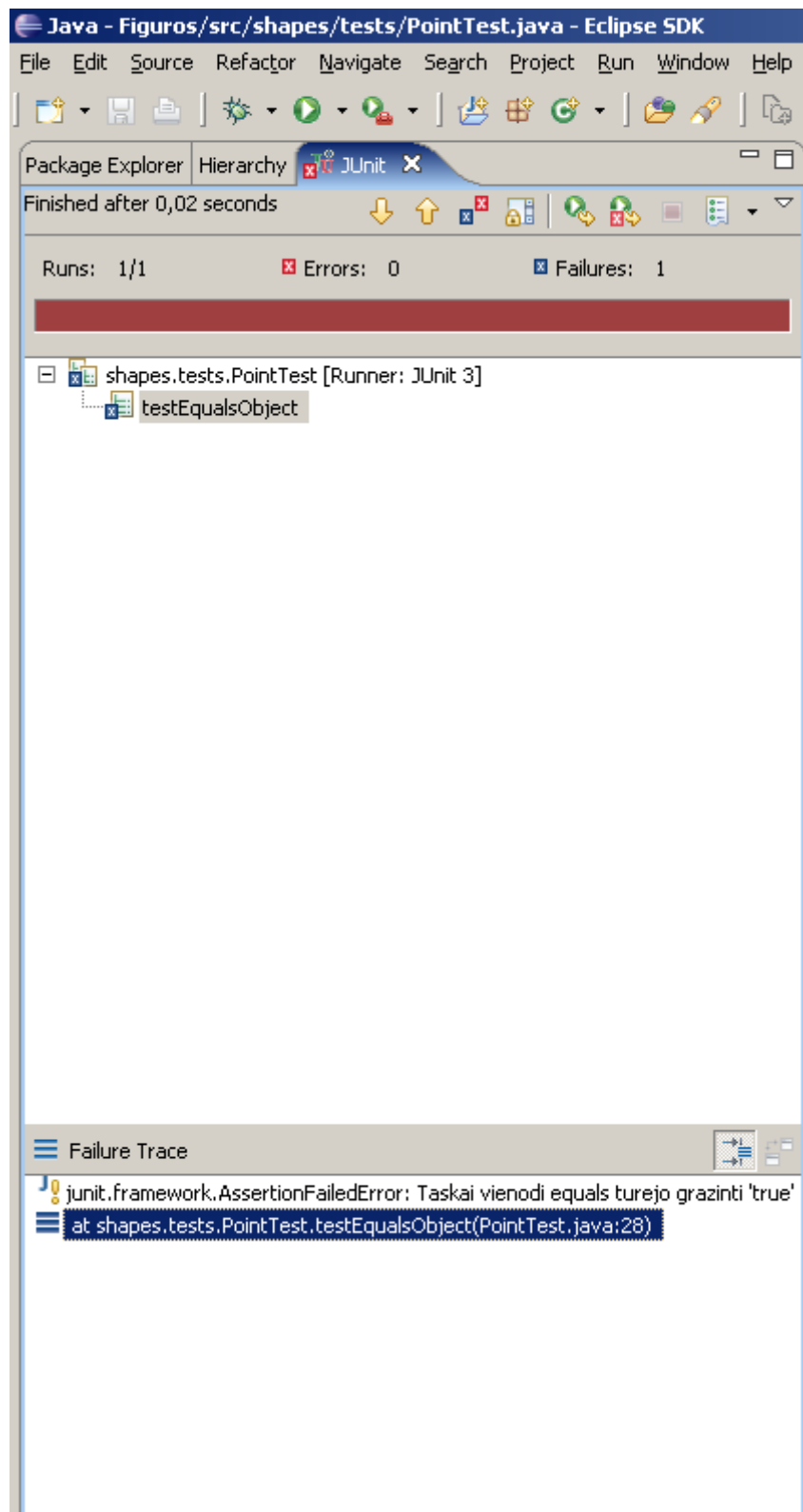


Pav. 21: Eclipse siūlomi sprendimai klaidai ištaisyti

Realizuoto testo paleidimas parodytas paveiksle 22. Testo rezultatai bus pavaizduoti Eclipse aplinkoje JUnit įskiepio kortelėje (Pav. 23). Raudona juosta viršuje rodo, kad testo nepavyko sėkmingai užbaigti - t.y. testuojamo metodo *equals* grąžinamas rezultatas nesutampa su laukiamu rezultatu. JUnit kortelės viduryje pateiktas vykdomų testų sąrašas, apačioje - testo nutraukimo priežastis. Pele spragtelėjus priežasties aprašymą bus parodyta klaidos vieta teste (Pav. 24). Pirmą ištaisykite klaidą klasės *Point* konstruktoriuje (Pav. 25). Ištaise klaidą paleiskite testą iš naujo, paleidimui naudokite paleidimo mygtuką (Pav. 26). Bus parodytas naujas klaidos pranešimas (Pav. 27), kurio ištaisyimas parodytas paveiksle 28. Ištaise šią klaidą ir paleidę testą iš naujo klaidų neturėtų likti ir JUnit kortelėje bus parodyta žalia juosta žyminti sėkmingą testo pabaigą (Pav. 29).



Pav. 22: PointTest testo paleidimas



Pav. 23: JUnit testavimo karkaso įskiepio kortelė Eclipse aplinkoje

```
PointTest.java x Point.java
package shapes.tests;

import shapes.Point;
import junit.framework.TestCase;

public class PointTest extends TestCase {

    public void testEqualsObject() {

        int x = 3, y = 7;

        //sukuriam du vienuodus taskus
        Point point1 = new Point(x, y);
        Point point2 = new Point(x, y);

        //istestuojam ar metodas equals metodas grazina true kai taskai vienodi
        assertTrue("Taskai vienodi equals turejo grazinti 'true'", point1.equals(point2))

        //nustatom taskui point2 x koordinate didesne per viena
        point2.setX(x+1);

        //dabar taskai point1 ir point2 yra nebevienodi
        assertFalse("Taskai skirtingi, equals turejo grazinti 'false'", point1.equals(point2))

        //atstatom pradine x reiksme
        point2.setX(x);
        //istestuojam ar taskai vel vienodi
        assertTrue("Taskai vienodi equals turejo grazinti 'true'", point1.equals(point2))

        //dabar pakeiciam y koordinate
        point2.setY(y-1);

        //dabar taskai point1 ir point2 yra nebevienodi
        assertFalse("Taskai skirtingi, equals turejo grazinti 'false'", point1.equals(point2))

        //patikrinam ar tasku palyginimo metodas equals veikia su null tipo iejimo duome
        assertFalse("equals turejo grazinti 'false' kai lyginama su 'null'", point1.equals(null))

        //patikrinam ar tasku palyginimo metodas equals veikia su kitokio nei Point tipo
        assertFalse("equals turejo grazinti 'false' kai lyginama su 'String'", point1.equals(""));

    }
}
```

Pav. 24: Testo nutraukimo vieta

```

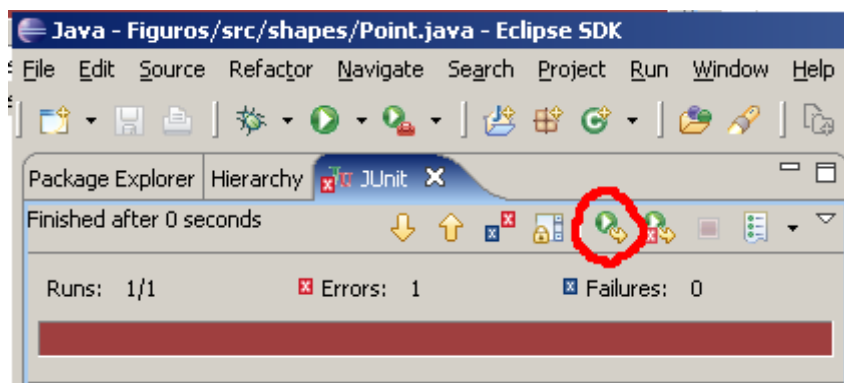
/**
 * Constructs point with specified X and Y coordinates.
 * @param x point X coordinate.
 * @param y point Y coordinate.
 */
public Point(int x, int y) {

    // incorrect implementation - assignment has no effect
    //x = x;
    //y = y;

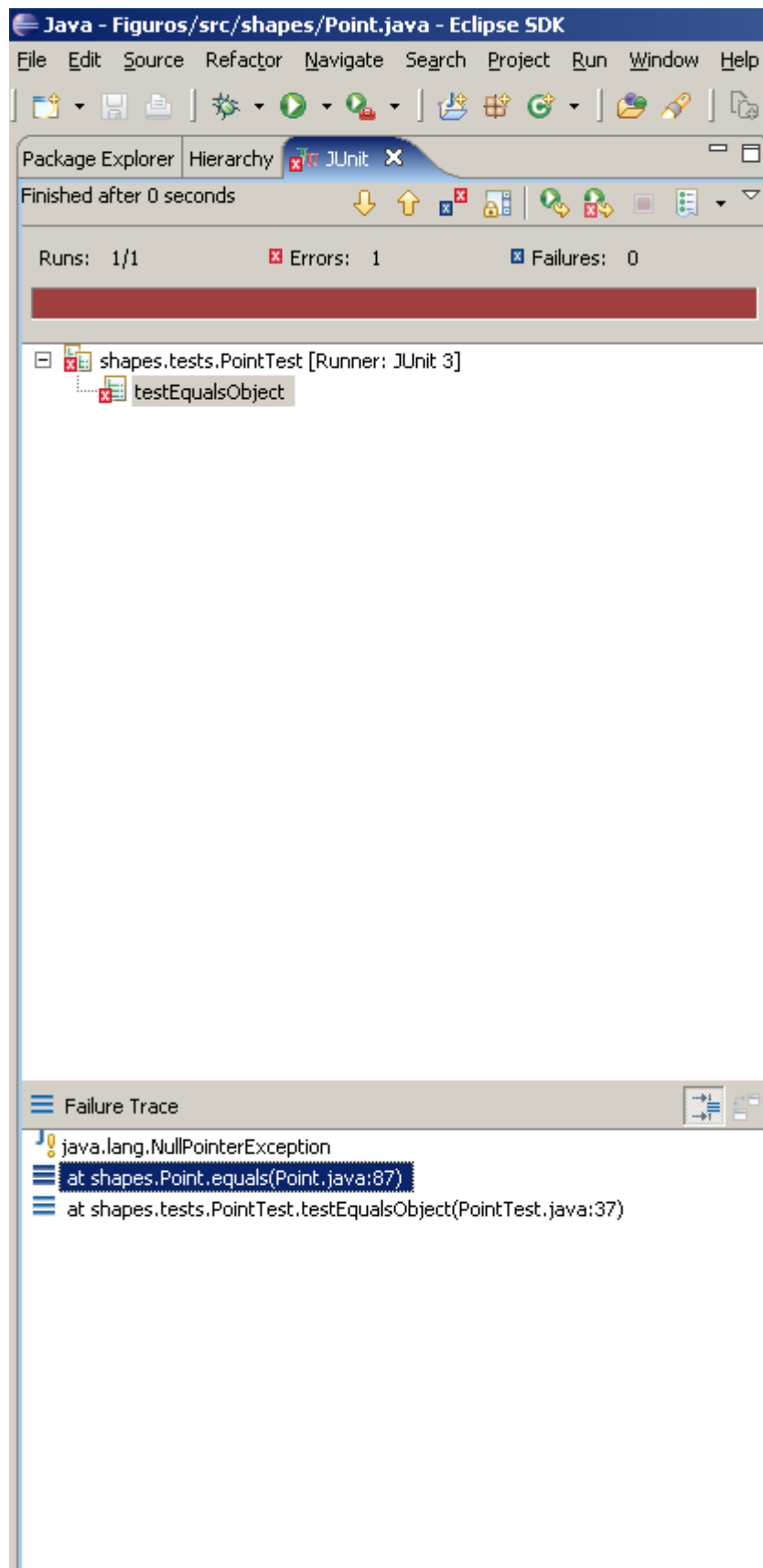
    // correct implementation
    this.x = x;
    this.y = y;
}

```

Pav. 25: Klaidos ištaisymas *Point* konstruktoriuje



Pav. 26: Mygtukas testo paleidimui



Pav. 27: Tuščio tipo klaida *equals* metode


```

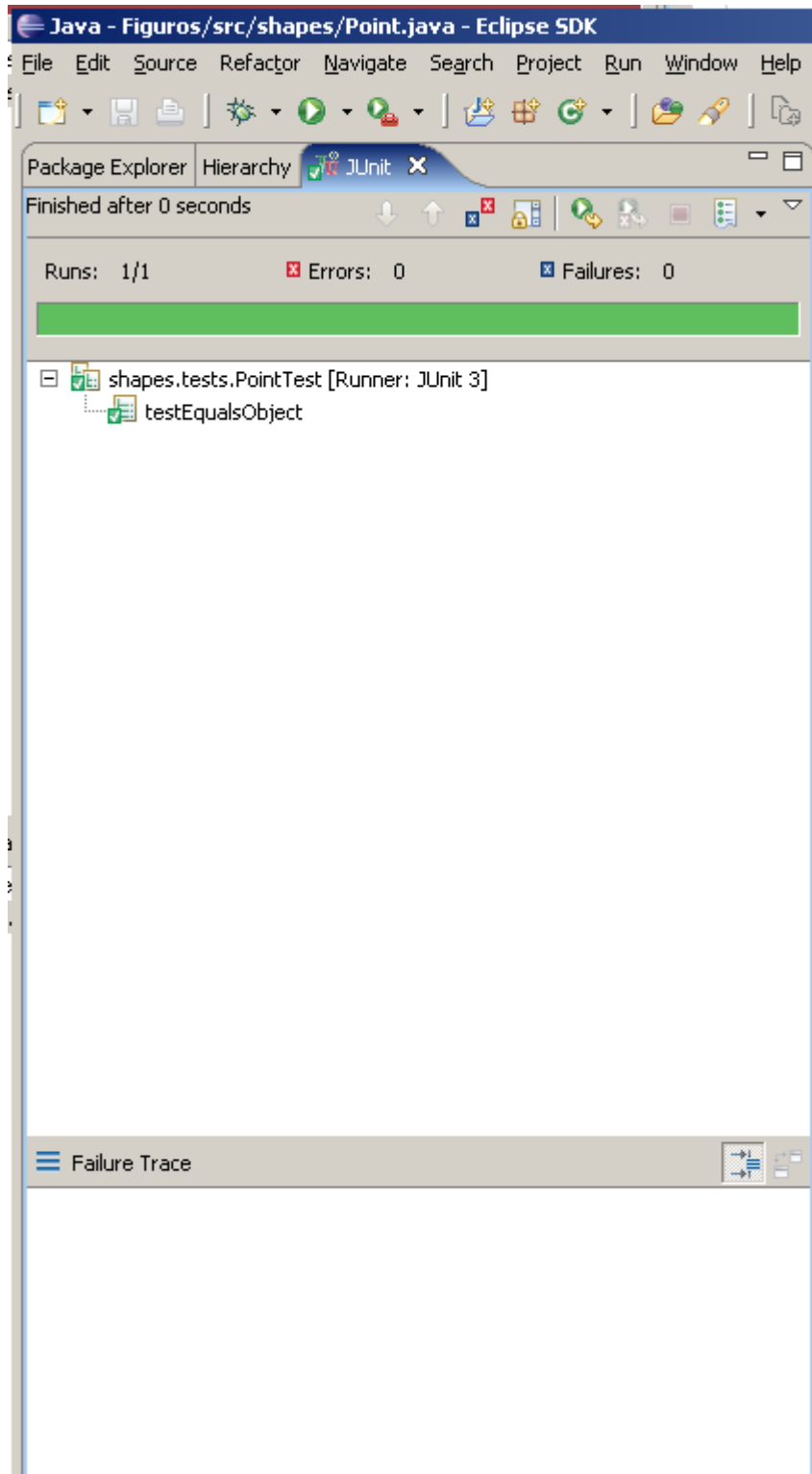
/**
 * Compares this point to other object.
 * Returns true only if passed object is also point and their X and Y coordinates a
 * false otherwise.
 * @param obj object to which this point should be compared.
 * @return true if passed object is point with the same X and Y coordinates,
 *         false otherwise.
 * @see java.lang.Object#equals(java.lang.Object)
 */
public boolean equals(Object obj) {

    // correct implementation don't forget to check for other types first
    if (!(obj instanceof Point)) {
        return false;
    }

    Point point = (Point) obj;
    return (this.x == point.x) && (this.y == point.y);
}

```

Pav. 28: Tipų nesuderinamumo klaidos ištaisymas



Pav. 29: Sėkminga testo pabaiga