

11 Prvá desktopová aplikácia

11.1 Desktopová aplikácia

Všetky aplikácie, ktoré sme doteraz vytvárali, boli tzv. konzolové aplikácie. To znamená, že vstupy a aj výstupy boli uskutočňované cez systémovú konzolu, v prípade vstupov to bola klávesnica a v prípade výstupov, monitor. V NetBeans IDE sme vstupy zadávali a výstupy sa nám zobrazovali v paneli **Output**, v spodnej časti obrazovky.

Konzolové aplikácie však vôbec nie sú pre Javu typické. Prevažná väčšina programov využíva tzv. **grafické rozhranie**, s množstvom vizuálnych komponent – formuláre, tlačidlá, prepínače, dialógy atď. Tieto komponenty tvoria grafické rozhranie, ktoré uľahčuje komunikáciu používateľa s aplikáciou.

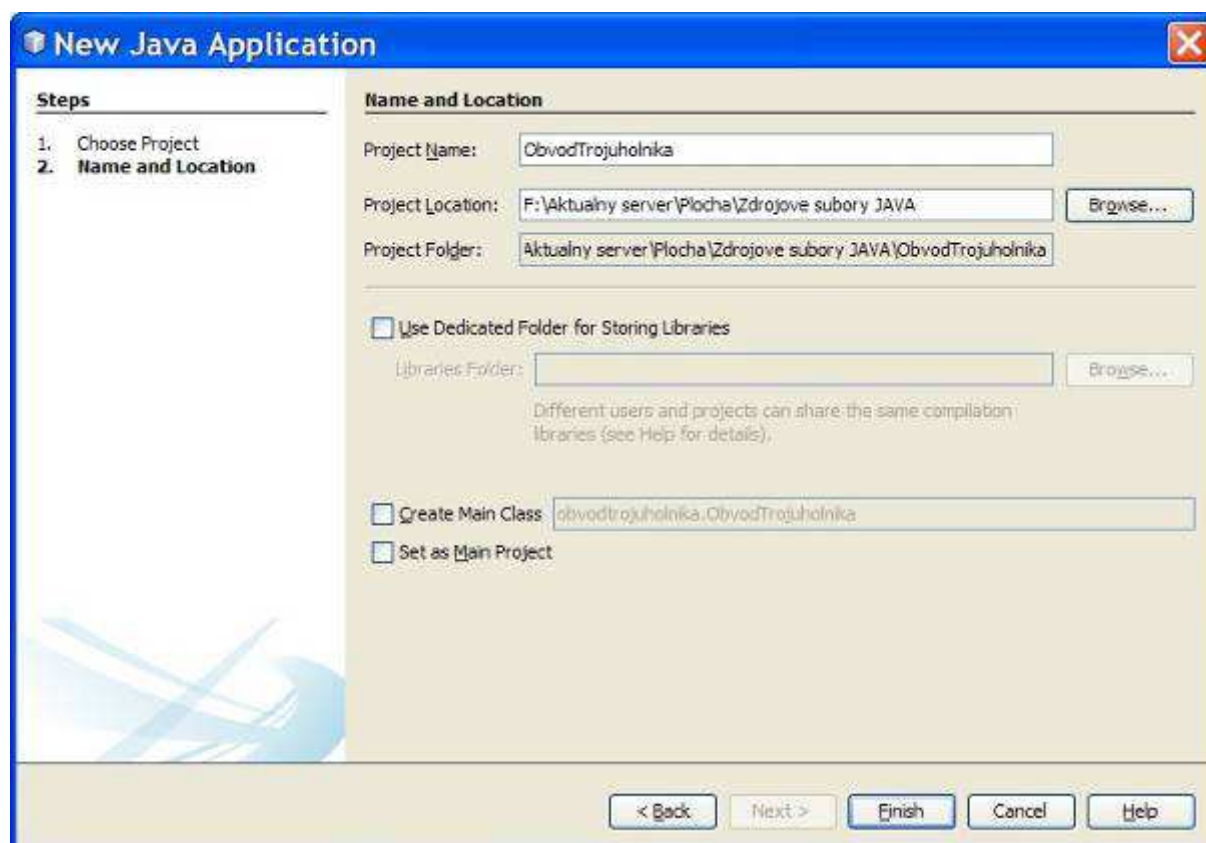
Aplikácie s grafickým rozhraním nazývame **desktopové aplikácie**.

Podrobnejšie sa s problematikou grafického rozhrania Javy budeme zaoberať neskôr. Teraz si vytvoríme prvú desktopovú aplikáciu, ktorá nám odhalí ďalšiu stránku Javy a zároveň bude (dúfajme) aj motiváciou do ďalšieho štúdia Javy.

11.2 Desktopová aplikácia ObvodTrojuholnika

Vytvoríme desktopovú aplikáciu s názvom **ObvodTrojuholnika**. Postupovať budeme rovnako ako pri vytváraní konzolovej aplikácie, len v okne **New Java Application** zrušíme zaškrtnutie políček **Create Main Class** a **Set as Main Project**. Potom zatlačíme **Finish**.

Obr.1

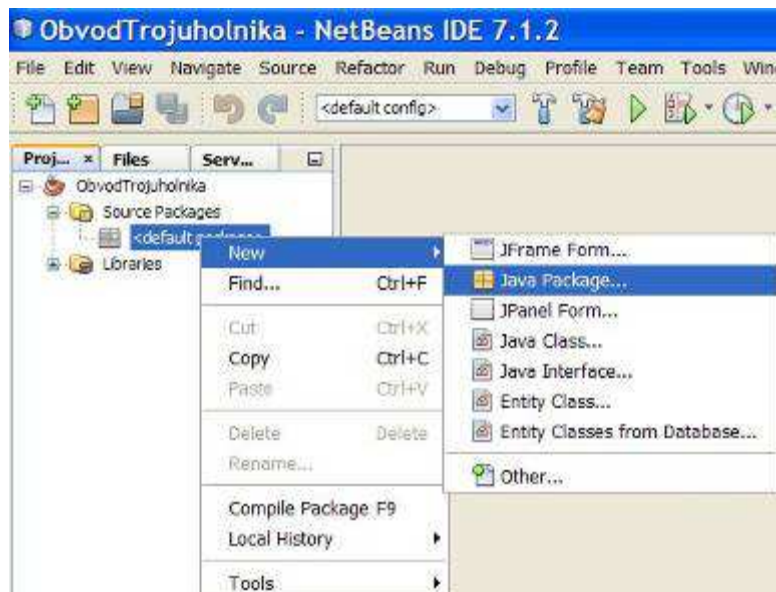


11 Prvá desktopová aplikácia

Otvorí sa nám prázdna aplikácia, ktorá neobsahuje ani balík. Ten si vytvoríme nasledovne:

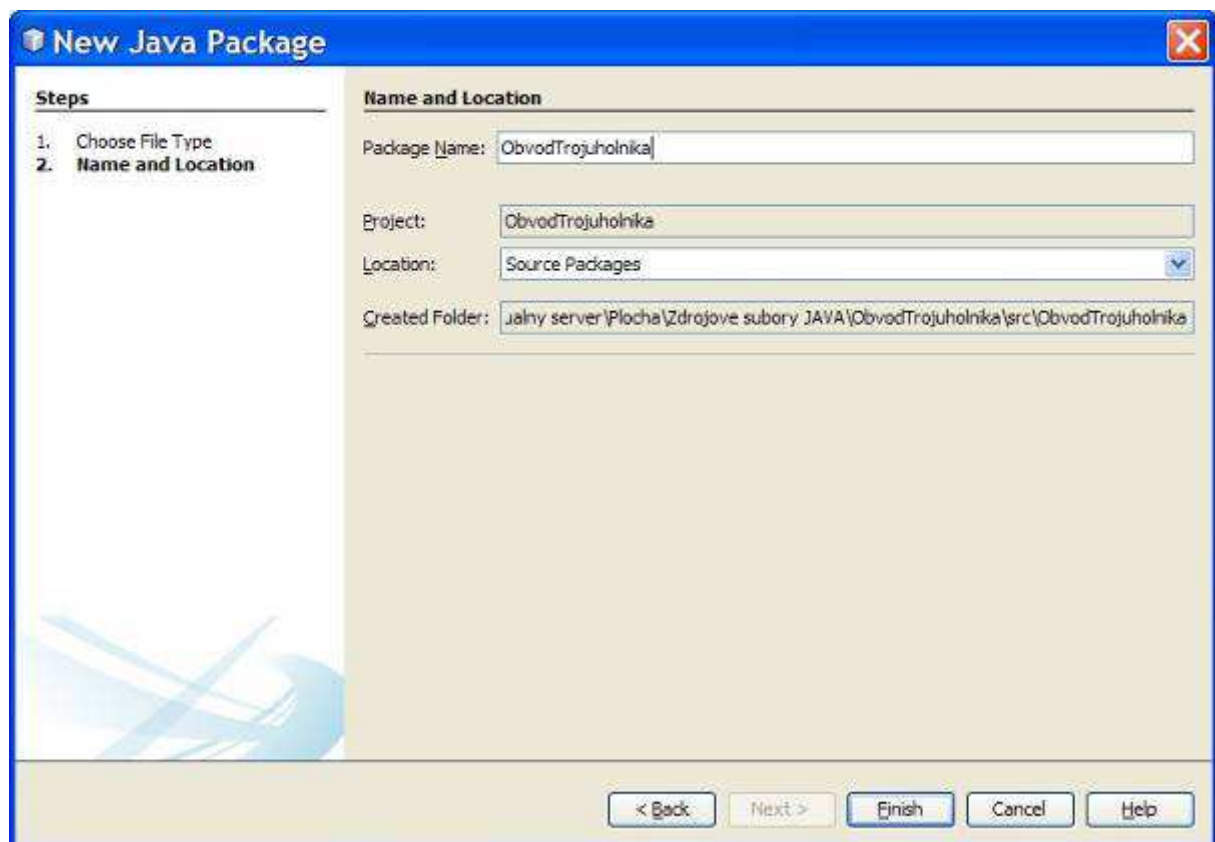
1. Rozbalíme strom projektu a klikneme pravým tlačidlom myši na **<default package>**
2. Postupne zvolíme **New/Java Package...**

Obr.2



V ponúknutom okne zapíšeme názov balíka **ObvodTrojuholnika** a zatlačíme **Finish**.

Obr.3

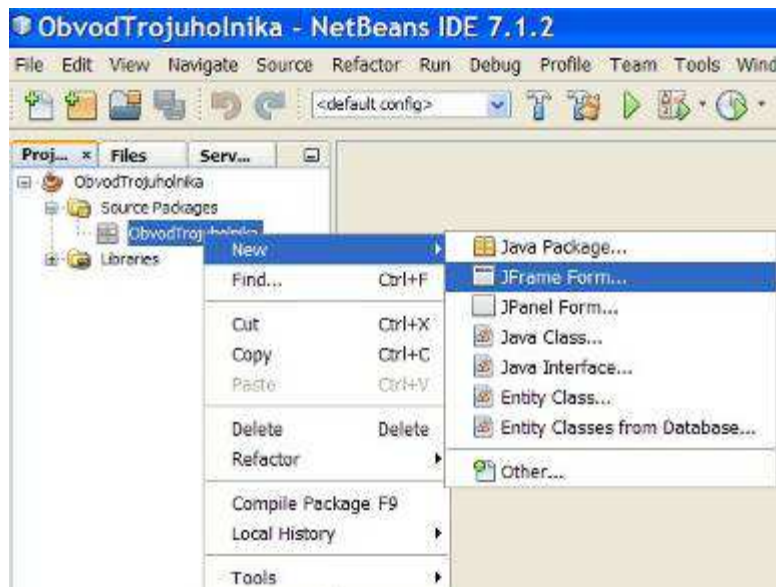


11 Prvá desktopová aplikácia

Teraz prikróčime k vytvoreniu samotného okna, ktoré sa po spustení aplikácie otvorí. Cez toto okno budeme komunikovať s našou aplikáciou.

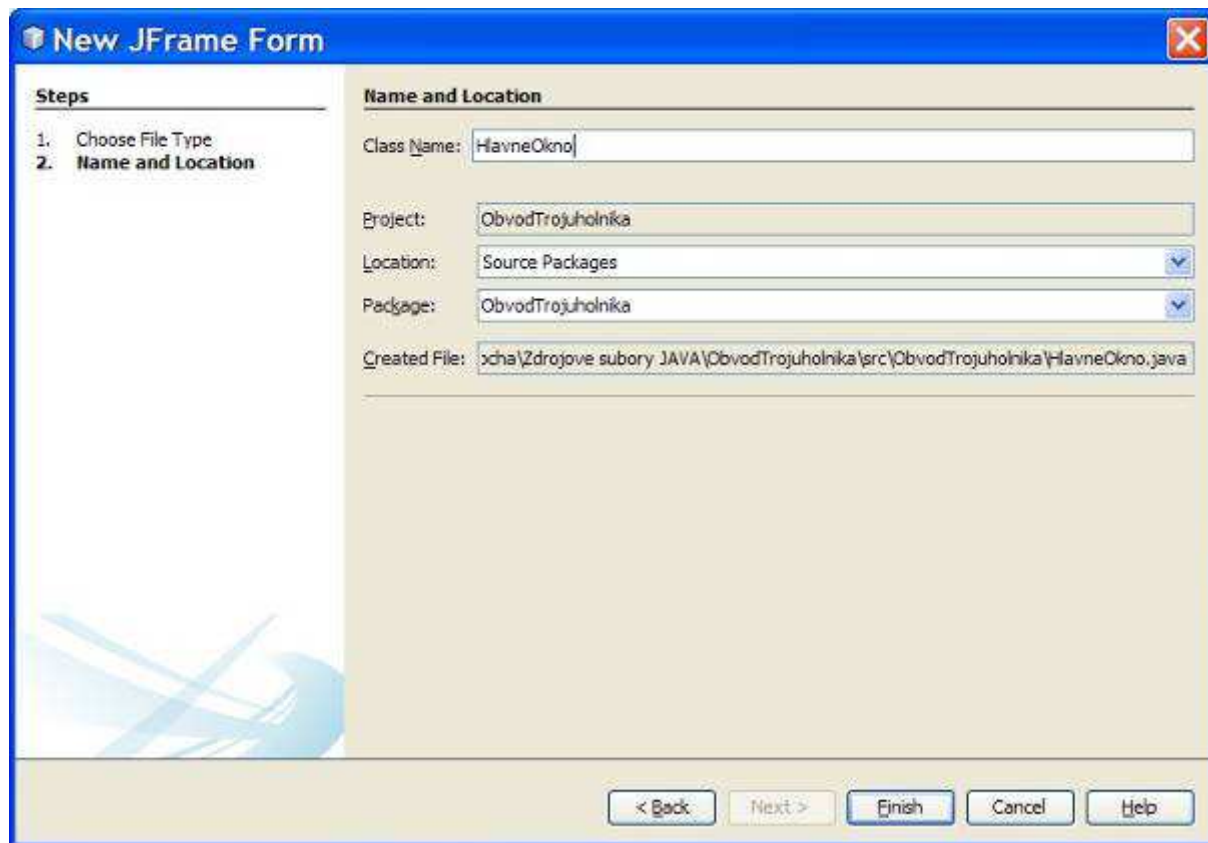
1. V strome projektu klikneme pravým tlačidlom myši na balík **ObvodTrojuholnika**
2. Postupne zvolíme **New/JFrame Form...**

Obr.4



V ponúknutom okne zapíšeme názov **HlavneOkno** a zatlačíme **Finish**.

Obr. 5



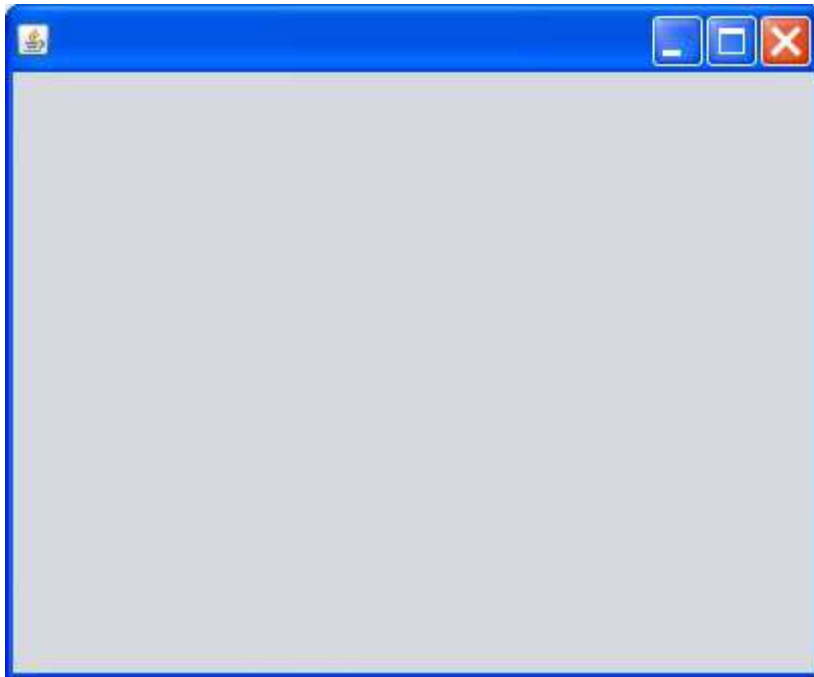
Týmto postupom sme vytvorili okno, ktoré sa nám otvorí po spustení aplikácie. Aplikáciu môžeme hneď spustiť rovnako ako sme spúšťali ktorúkoľvek konzolovú aplikáciu.

11 Prvá desktopová aplikácia

NetBeans IDE si ešte vyžiada vytvorenie hlavnej triedy – v ponúknutom okne zatlačíme **OK**.

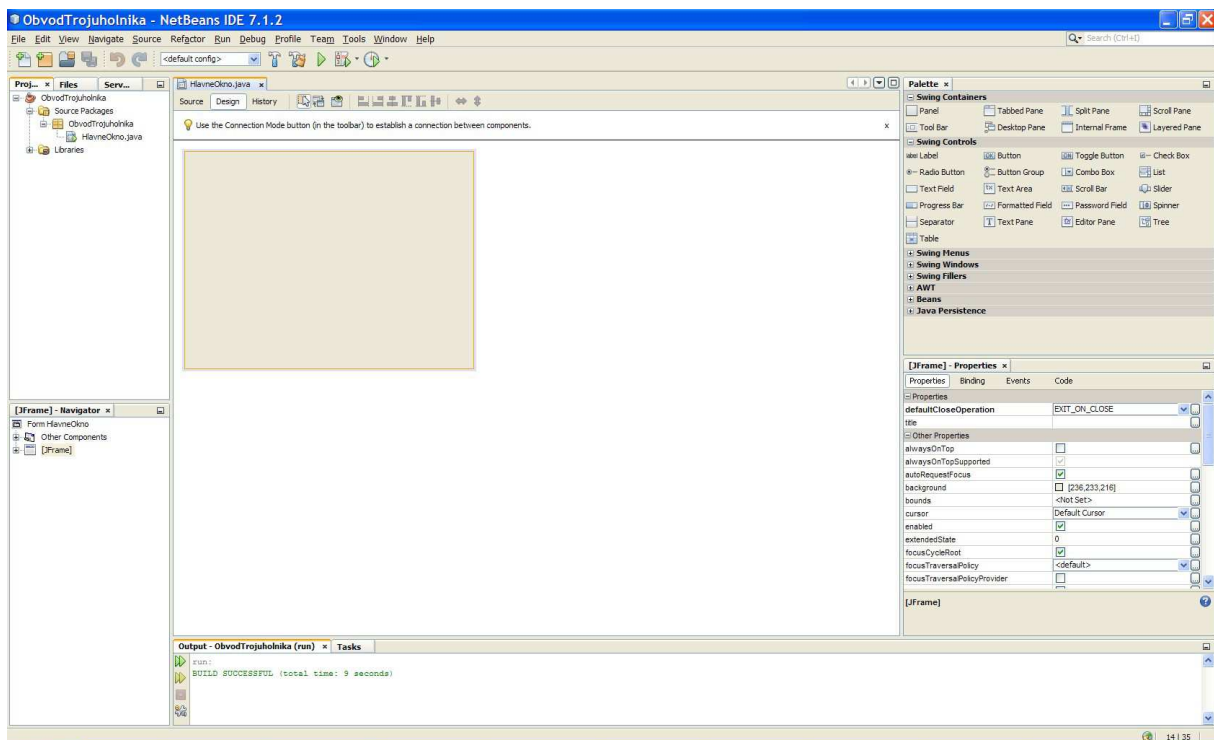
A toto je náš výsledok:

Obr.6



Vidíme, že okno je prázdne, v titulnom páse nemá žiadny názov, ale obsahuje klasické tlačidlá v pravej časti titulného pásu. Okno zatvorme a pozrime sa ako vyzerá prostredie NetBeans IDE.

Obr.7



11 Prvá desktopová aplikácia

V paneli, kde sa v konzolových aplikáciách zvyčajne zobrazoval zdrojový kód, máme teraz zobrazený dizajn nášho desktopového okna (záložka **Design**). Prostredie NetBeans IDE vygenerovalo aj množstvo zdrojového kódu, ktorý si môžeme zobraziť, ak klikneme na záložku **Source**. Nebudeme sa teraz zaoberať podrobnosťami ohľadne zdrojového kódu, len si všimneme dôležitú skutočnosť a to, že v zdrojovom kóde sa nachádza metóda **main()**.

Ďalej si všimnime, že v pravej časti vývojového prostredia (musíme byť v záložke Design) sa nachádza ponuka grafických komponent – **Palette** (v hornej časti) a ich vlastnosti spolu s udalosťami, ktoré generujú **[JFrame] – Properties** (v spodnej časti obrazovky).

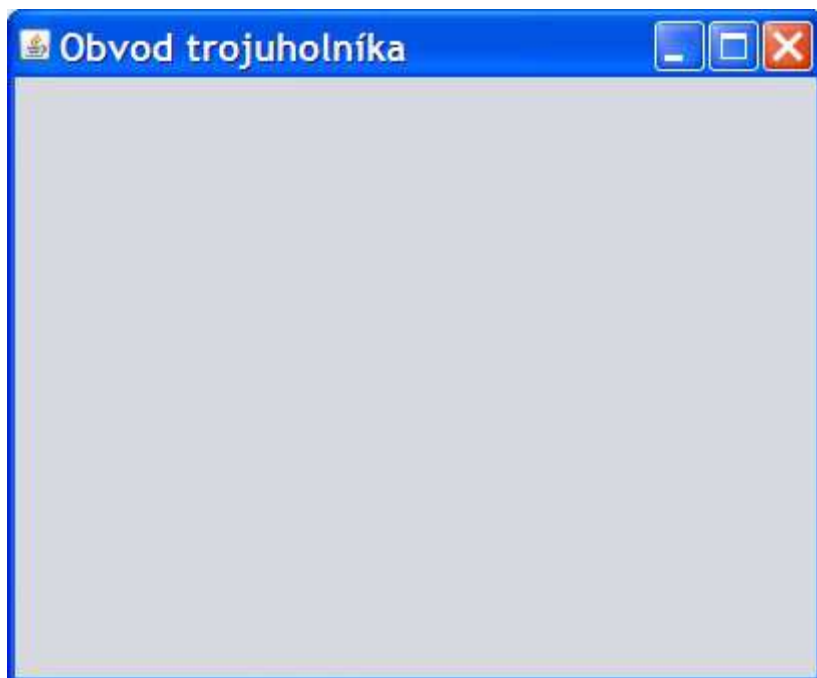
Jednu z vlastností teraz hneď aj zmeníme. Vlastnosťou **title** nastavíme nášmu oknu názov. Zapišme názov *Obvod trojuholníka*.

Obr.8



Naše okno teraz vyzerá nasledovne:


Obr.9




11 Prvá desktopová aplikácia

Teraz môžeme prísť k umiestneniu potrebných komponent do nášho okna. V desktopových aplikáciách budeme, podobne ako aj v konzolových aplikáciách, realizovať vstupy a výstupy. Preto budeme potrebovať komponenty, cez ktoré budeme zadávať vstupné údaje a komponenty, ktoré nám budú zobrazovať výsledky.

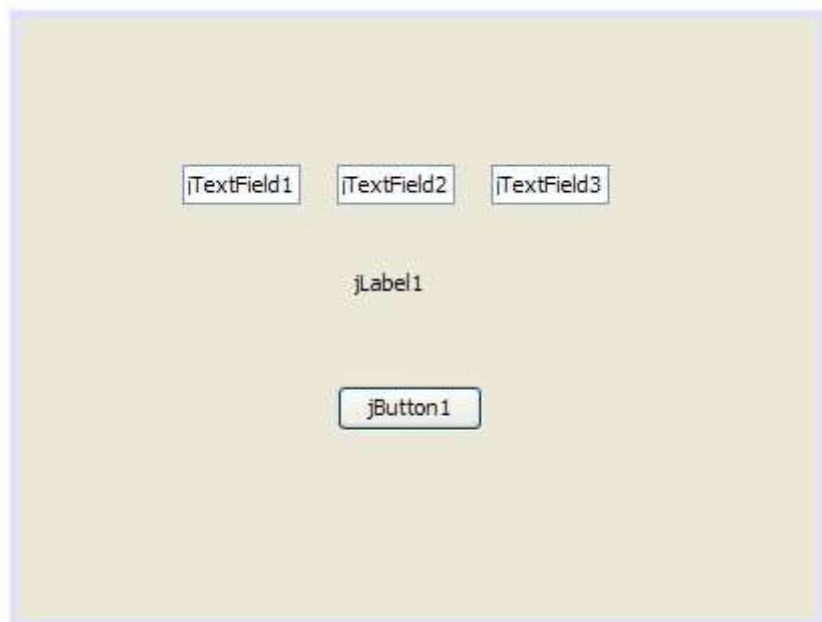
Na vstup údajov nám posluží komponent **Text Field**  Text Field

Na výstup údajov použijeme komponent **Label**  label Label

Desktopové aplikácie bežia v operačnom systéme inak ako konzolové. Veľmi jednoducho možno povedať (podrobnejšie sa s behom desktopových aplikácií budeme zaoberať neskôr), že ide vlastne o zacyklené programy, ktoré čakajú na reakciu používateľa – kliknutie myšou, zatlačenie klávesu, zápis vstupnej hodnoty a pod. Preto musíme aj v našej aplikácii použiť komponentu, po zatlačení ktorej sa vstupné údaje, zapísané v textových poliach, načítajú do premenných, vypočíta sa obvod trojuholníka a výsledok sa zobrazí v komponente Label. Na vykonanie spomínaných operácií použijeme komponent **Button**  Button (v skutočnosti obslúžime udalosť **actionPerformed**, ale o tom neskôr).

Všetky spomínané komponenty teraz umiestnime do okna. Potrebujeme tri komponenty Text Field, jeden komponent Label a jeden komponent Button. Premiestnenie urobíme nasledovne: klikneme v palete na príslušný komponent a následne klikneme na formulár. Výsledok by mohol vyzeráť napríklad tak, ako to zobrazuje nasledovný obrázok:

Obr.10



Rozmiestnenie jednotlivých komponent môžeme kedykoľvek zmeniť, rôznym spôsob zarovnať atď.

Keďže po spustení aplikácie sa zobrazujú názvy jednotlivých komponent, zmeníme niektoré ich vlastnosti. Pri zmene vlastností komponent musíme dávať pozor na to, aby ten komponent, ktorého vlastnosť meníme, bol aktívny.

11 Prvá desktopová aplikácia

Najskôr zmeníme vlastnosť **text** komponentu `jButton1` (všimnite si, že keď nejaký komponent premiestnime na formulár, tak sa upraví jeho meno – prečo je to tak si vysvetlíme, keď budeme preberať objektovo orientované programovanie): k vlastnosti **text** zapíšme *Výpočet*.

Komponentom `jTextField1` , `jTextField2` a `jTextField3` text vo vlastnosti **text** zmažeme. Potom všetkým trom komponentom nastavíme vlastnosť **HorizontalSize** na **60**.

Komponentu `jLabel1` text vo vlastnosti **text** tiež zmažeme.

Pre všetky komponenty by sme mohli meniť ďalšie vlastnosti, napr. zarovnanie zobrazovaného textu, fonty, farbu pozadia atď.

Teraz, keď spustíme aplikáciu, by mala vyzeráť nasledovne.

Obr.11



Do textových polí môžeme zapisovať hodnoty, klikať na tlačidlo *Výpočet*, presúvať okno, meniť jeho veľkosť, jednoducho používať aplikáciu. Tá však stále nerobí to, kvôli čomu sme ju v skutočnosti vytvorili. My chceme, aby počítala obvod trojuholníka. Preto teraz musíme napísať kód, ktorý sa má vykonať, keď na tlačidlo *Výpočet* klikneme myšou.

Zatvorme našu aplikáciu a v NetBeans IDE si zobrazme formulár s komponentmi (čiže ak nie sme, tak prejdeme na kartu Design).

Potrebujeme teraz, aby `jButton1` (t.j. naše tlačidlo *Design*) reagoval na kliknutie myšou. Vo formulári dvakrát klikneme na tlačidlo `jButton1` , čím prejdeme do zdrojového kódu. Vývojové prostredie nám vygeneruje prázdnu metódu

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
```

Táto metóda sa zavolá práve vtedy, keď klikneme na tlačidlo *Výpočet* pri spustenej aplikácii.

Do tela metódy musíme teraz zapísať príkazy, ktoré nám prečítajú hodnoty z textových polí, prekonvertujú ich na čísla, tieto čísla spočítajú a následne v komponente `jLabel1` vypíšu výsledok.

11 Prvá desktopová aplikácia

Napíšme preto nasledovné príkazy:

```
String sStranaA = jTextField1.getText();
String sStranaB = jTextField2.getText();
String sStranaC = jTextField3.getText();
int stranaA = Integer.parseInt(sStranaA);
int stranaB = Integer.parseInt(sStranaB);
int stranaC = Integer.parseInt(sStranaC);
int vysledok = stranaA + stranaB + stranaC;
String sVysledok = Integer.toString(vysledok);
jLabel1.setText(sVysledok);
```

Stručné vysvetlenie kódu: na prvých troch riadkoch sú vytvárané reťazce `sStranaA`, `sStranaB`, `sStranaC` a sú do nich načítané hodnoty, ktoré zapisujeme do textových polí, po spustení aplikácie (Pozor! Všetky hodnoty zapisované do textových polí sú reťazce). Na ďalších troch riadkoch sú vytvárané celočíselné premenné, do ktorých sú uložené prekonvertované hodnoty reťazcov. Ďalej je vytvorená premenná `vysledok` a do nej uložený súčet strán trojuholníka. Hodnota v premennej `vysledok` je potom konvertovaná na reťazec na uložená do reťazca `sVysledok`. Ten je potom zobrazený vo vlastnosti `text` komponentu `jLabel1`.

Teraz môžeme spustiť našu aplikáciu, zapísať veľkosť strán trojuholníka a kliknutím na *Výpočet*, nechať vypočítať a zobrazíť obvod trojuholníka.

Obr.12



11 Prvá desktopová aplikácia

Ukončili sme našu prvú desktopovú aplikáciu. Mohli by sme v nej vykonať ešte množstvo zmien, napríklad v okne aplikácie zobrazíť názvy jednotlivých strán trojuholníka a pod.

Naším cieľom však bolo predovšetkým ukázať, ako sa tvoria aplikácie s grafickým rozhraním. Videli sme, že vytvorenie aj algoritmicky jednoduchšej úlohy je časovo náročnejšie, než vytvorenie konzolovej aplikácie. Výsledok je ale veľmi rozdielny. Aplikácia s grafickým rozhraním je používateľsky oveľa prívetivejšia a v dnešnej dobe prakticky nevyhnutná. Preto aj v budúcnosti budeme niektoré aplikácie vytvárať ako desktopové až kým definitívne neprejdeme k štúdiu samotného rozhrania **SWING**, ktorému vďačíme za všetko to, čo sme dnes vytvorili.

11.3 Cvičenie

1. Vytvorte desktopovú aplikáciu s názvom **KvadratickaRovnica**. Program vypočíta korene kvadratickej rovnice. V prípade, že neexistuje riešenie kvadratickej rovnice v oblasti reálnych čísel, tak to program vypíše.
2. Vytvorte desktopovú aplikáciu s názvom **ObsahPlastaValca**. Vo vytvorenej aplikácii zapíšete polomer podstavy a výšku valca. Po kliknutí na tlačidlo výpočet sa zobrazí obsah plášťa valca.
3. Vytvorte desktopovú aplikáciu s názvom **PrevodStupnovNaRadiany**. Po zadaní uhla v stupňoch program vypočíta a zobrazí hodnotu v radiánoch.

11.4 Otázky

1. Aký je rozdiel medzi desktopovou a konzolovou aplikáciou?
2. Ktorý komponent používame na vstup hodnôt?
3. Ktorý komponent používame na zobrazenie výsledkov?
4. Ktoré vlastnosti komponent sme využili pri tvorbe aplikácie ObvodTrojuholníka?
5. Ktorý komponent a ktorú udalosť sme využili pre výpočet a zobrazenie výsledku?