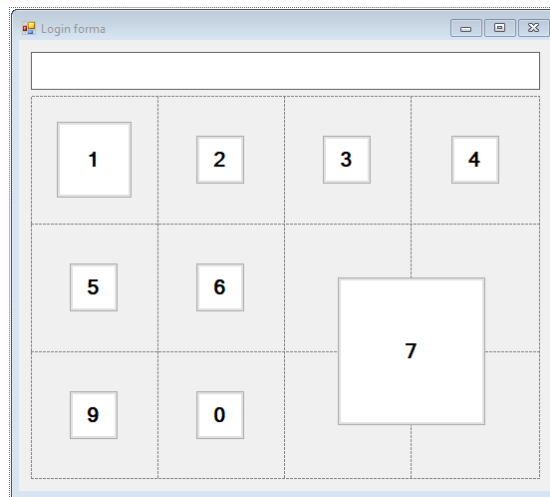


ZADATAK

U zadatku TMP_2 potrebno je napraviti formu kao na slici.



The screenshot shows a Java Swing window titled "Login forma". Inside the window, there is a text input field at the top. Below it, there is a grid of buttons arranged in a 3x4 layout. The buttons are labeled with numbers 1 through 9, and a larger button labeled 7. The buttons are arranged as follows:

1	2	3	4
5	6	7	
9	0		

U zadatku je potrebo omogućiti da se pritiskom na bilo koji od tastera u tekstualnom polju upiše broj koji je pritisnut. Cilj zadatka je da obezbjedi funkcionalnost sa što manje linija koda. Kontrole na formi rasporediti korištenjem `TableLayoutPanel`-a. Cilj ove vježbe je da se u potpunosti savlada `TableLayoutPanel`-a i da se upoznaju atributi koji su bitni za rad sa njim. Takođe u okviru ove vježbe je objašnjeno kako je moguće jednu te istu metodu koristiti za više tastera na formi i kako je moguće "pročitati" koji je taster pritisnut na osnovu argumenata metode.

RJEŠENJE I OBJAŠNJENJE KODA

Dizajniranje forme

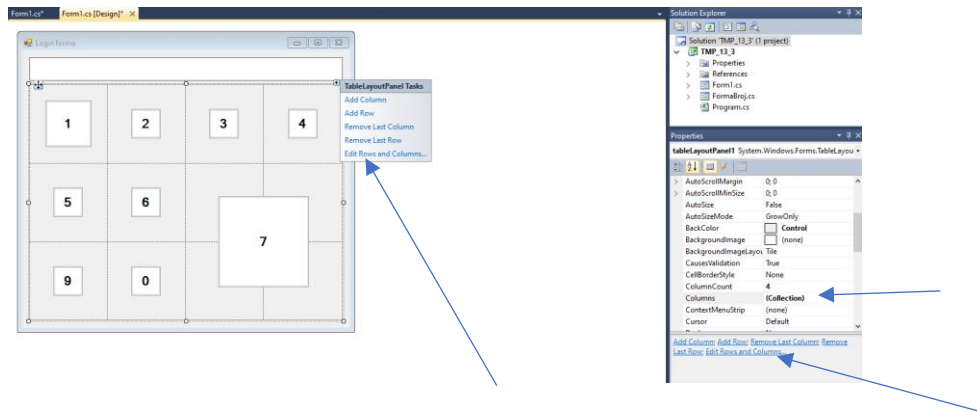
Na formi je potrebno postaviti jedan TextBox u kome će se ispisivati brojevi koje je korisnik pritisnuo.

Na formi je potrebno postaviti jedan TableLayoutPanel sa 4 kolone i 3 reda, i pozicionirati ga ispod TextBox-a, kao što je prikazano na prethodnoj slici. Potrebo je podesiti TableLayoutPanel tako da sve ćelije budu iste veličine.

Rasporediti tastere kao na slici i pogledati koje attribute ima TableLayoutPanel.

TableLayoutPanel je kontrola koja se koristi da bi se elementi na formi skladno rasporedili i da bi svoju veličinu i raspored mjenjali u skladu sa promjenom veličine forme.

Kolone i redovi TableLayoutPanel-a se dodaju i podešavaju u prozoru **Column i Row Style**. Ovaj prozor moguće je otvoriti na više načina, što je prikazano na slici ispod.

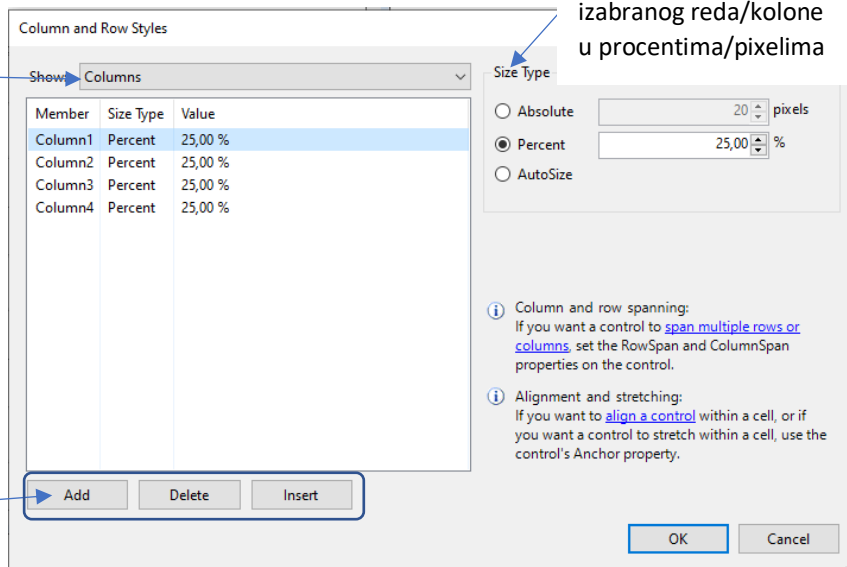


Prozor za podešavanje panela prikazan je na slici ispod.

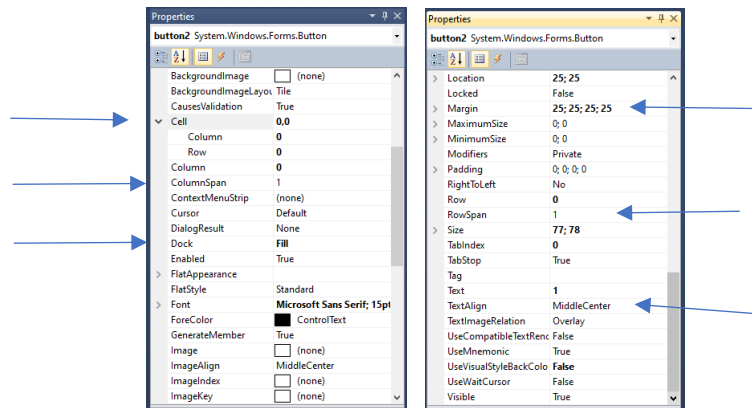
Podešava se veličina izabranog reda/kolone u procentima/pixelima

Bira se da li se podešavaju redovi ili kolone

Dodaju se, brišu ili ubacju novi redovi/kolone



Kod TableLayoutPanela u jednu ćeliju moguće je ubaciti samo jednu kontrolu. Vrijednost atributa Anchor potrebno je podesiti na Top, Bottom, Left, Right, čime će se omogućiti da TableLayoutPanel mijenja svoju veličinu sa promjenom veličine forme. Kada se neka kontrola nalazi u TableLayoutPanelu onda se u dijelu Properties pojavljaju neki novi atributi koje je moguće mjenjati.



Obratiti pažnju na označene attribute i pratiti šta se dešava njihovim mjenjanjem.

Cell – U polju cell nalazi se redni broj kolone i reda u kojoj se kontrola nalazi.

ColumnSpan – Označava preko koliko kolona se prostire kontrola. Kod broja 7 vrijednost ovog atributa je 2, što označava da se taj taster prostire preko 2 kolone.

Dock – Ovaj atribut omogućava da se kontrola veže za roditeljsku kontrolu (u ovom primjeru je to TableLayoutPanel). Isprobati sve opcije kako bi vidjeli kako kontrola reguje na promjenu ovog atributa.

Margin - Ovdje može da se postavi udaljenost kontrole od ivica TableLayoutPanela.

RowSpan - Označava preko koliko redova se prostire kontrola. Kod broja 7 vrijednost ovog atributa je 2, što označava da se taj taster prostire preko 2 reda.

TextAlign – Omogućava da se promjeni poravnanje teksta na samoj kontroli.

Kodiranje aplikacije

U okviru ovog primjera potrebno je da u TextBox txtPolje upišemo ono što piše na tasteru unutar atributa Text (Ukoliko se pritisne 1 u polje se upiše broj 1, ukoliko se pritisne 2 u polje se upisuje broj 2 i tako redom). Na osnovu prethodno stečenog znanja kao prvo rješenje nameće se da svaki taster kreiramo metodu koja će u polje upisati Text sa tastera. Dio koda koji bi imao ovu funkcionalnost prikazan je ispod:

```
private void button2_Click(object sender, EventArgs e)
{
    txtPolje.AppendText(button2.Text);
}

private void button3_Click(object sender, EventArgs e)
{
    txtPolje.AppendText(button3.Text);
}
```

Za dodavanje teksta u TextBox korištena je funkcija AppendText koja automatski dodaje neki string na postojeći sadržaj u TextBox-u. Nešto primitivniji način da se ovo uradi je:

```
txtPolje.Text += button3.Text;
```

Rješavanje zadatka na ovaj način bi zahtjevalo da za svaki taster imamo posebnu metodu i u svakoj metodi mora se voditi računa o tome koji je taster pritisnut, koje je njegovo ime, a i dobili bi mnogo mnogo linija koda koji se ponavlja.

Da bi se na najbolji način riješio ovaj problem potrebno je kreirati jednu metodu, ovdje se ta metoda zove **click** (Ime je moglo biti proizvoljno), koja predstavlja događaj – (što znači da od argumenata ima promjenljivu **sender** koja je tipa **object** i promjenljivu **e** tipa **EventArgs**). Može se vidjeti da je konstrukcija ove metode ista kao i gore kreirane **button2_click** i **button3_click**. Unutar ove metode u prvom koraku definisana je jedna promjenljiva **btn**, koja je tipa **Button** (jer je korisnik kliknuo na Button) koji uzima vrijednost od sender-a, i pretvara je u tip Button. Sender predstavlja objekat koji je pokrenuo ovu akciju. Pošto se radi o događaju klik miša na Button, sender je Button.

```
private void click(object sender, EventArgs e)
{
    Button btn = (Button)sender;
    txtPolje.AppendText(btn.Text);
}
```

Linija koda `Button btn = (Button)sender;` se može napisati i na sledeći način: `Button btn = sender as Button;`

Ovom linijom koda se kreira taster btn koji predstavlja taster na koji je korisnik kliknuo i automatski preuzima sve njegove attribute. U narednom koraku u TextBox dodaje se tekst od tastera koji je korisnik kliknuo tako što se pozove funkcija `AppendText` nad TextBoxom `txtPolje`, što je objašnjeno ranije. Na ovaj način se desetine linija koda mjenja sa samo dvije linije koda.

Da bi ova metoda radila potrebno je da je pridružimo svakom tasteru koji je na formi. Na sledećoj slici je prikazano kako se ovo radi. Postupak je potrebno ponoviti za svaki taster.

