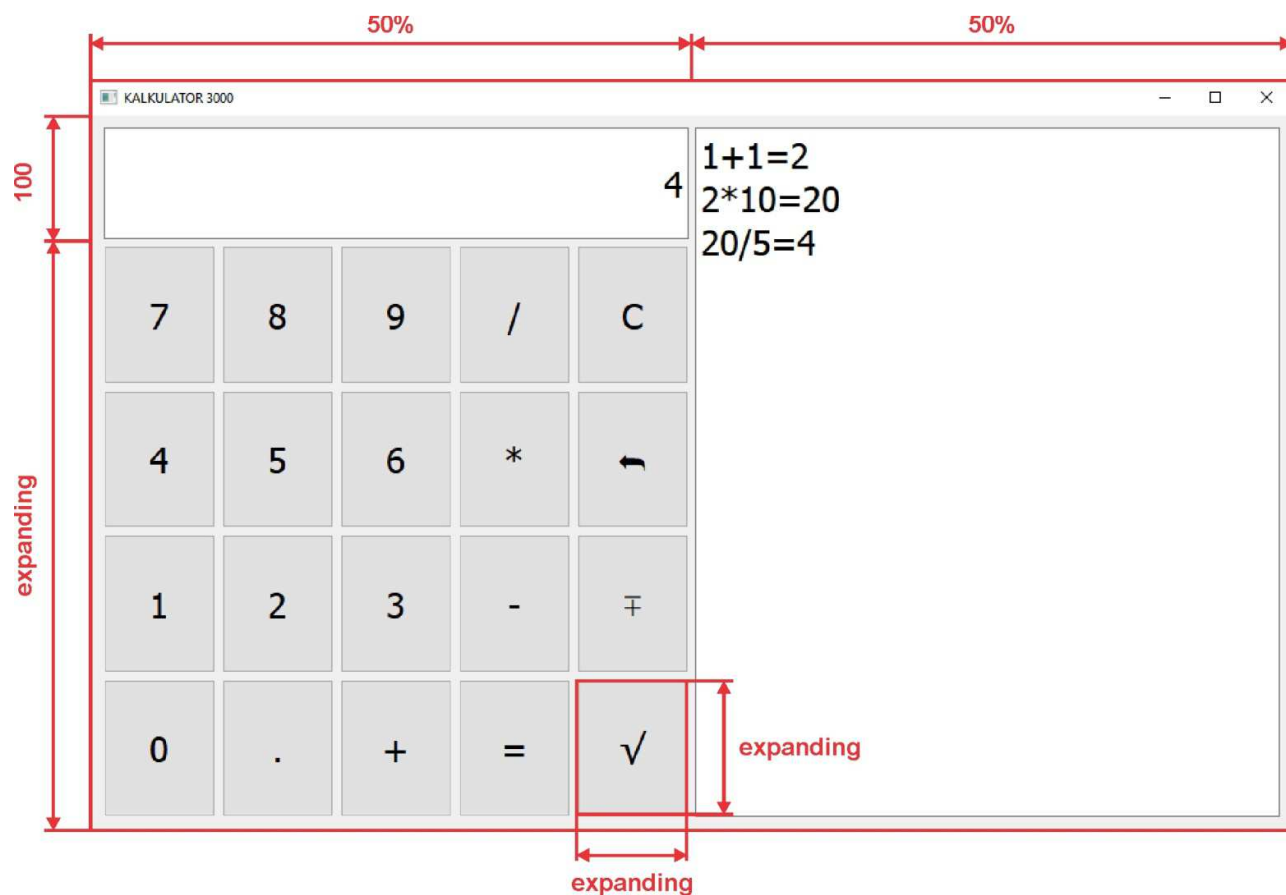


VEŽBA 5.



Cilj vežbe:

Kreiranje Qt Widget aplikacije. Raspoređivanje kontrola na formi korišćenjem *Layout* kontejnera. Prenos podataka korišćenjem signala i slotova.

Zadatak:

Potrebno je kreirati aplikaciju **Kalkulator** u okviru Qt razvojnog okruženja i implementirati sledeće funkcionalnosti:

1. Aplikacija treba da bude tipa **QWidget** i da se sastoji iz 3 dela: displej rezultata (gore levo - *LineEdit* kontrola), tastatura (dole levo), istorija izračunavanja (desno - *PlainTextEdit* kontrola). Za rapoređivanje kontrola na formi koristiti **Layout** kontejner kontrole u proporciji kao na slici. Prilikom promene širine i visine aplikacije kontrole moraju automatski proporcionalno da se povećavaju ili smanjuju. Svako od kontrola podesiti `objectName` parametar radi preglednosti koda.
2. Kreirati klasu **CalculatorLogic** (izvedenu iz **QObject**) u okviru koje će da bude implementirana logika rada kalkulatora. U okviru ove klase definisati signale **resultChanged(QString)** i **resultHistoryChanged(QString)**. Definisati metodu **doCommand(QString)** pomoću koje će se kalkulatoru prosleđivati komande (+, -, 0..9, ...). Ova

klasa će da čuva stanje trenutnog rezultata i u zavisnosti od toga koja komanda je prosledjena putem *doCommand* vršiće promenu rezultata i emitovaće signale *resultChanged* i *resultHistoryChanged*.

3. QWidget aplikacija treba da koristi instancu *CalculatorLogic* klase. U okviru ove aplikacije za svako dugme kalkulatora kreirati programski konekcije tako da povezuju signal *clicked* sa zajedničkom slot metodom *onBtnClicked()*. U okviru metode *onBtnClicked()* koristiti funkciju *sender()* kako bi odredili koje dugme je poslao signal. Nakon određivanja dugmeta koje je poslao signal prosledici instanci *CalculatorLogic* komandu pozivom metoda *doCommand*.

4. U okviru QWidget aplikacije kreirati slot metode *onResultChanged* i *onResultHistoryChanged* i programski kreirati konekcije sa signalima *resultChanged* i *resultHistoryChanged*. U okviru metoda *onResultChanged* ažurirati tekst u kontroli za prikaz rezultata a u okviru metoda *onResultHistoryChanged* ažurirati tekst u okviru kontrole za prikaz istorije izračunavanja.

5. Kalkulator treba da ima podršku za operacije: sabiranje, oduzimanje, množenje, deljenje i korenovanje kao i za funkcije: promena znaka, brisanje cifre i resetovanje.

Primer programskog toka podataka:

- Korisnik pritiska dugme 1, nakon čega dugme šalje signal *clicked* koji okida slot *onBtnClicked*. U okviru *onBtnClicked* se ispituje koje dugme je poslalo signal i nakon toga se poziva metod *doCommand* instance *CalculatorLogic* i prosleđuje "1". *CalculatorLogic* ažurira stanje rezultata i šalje signal *resultChanged* koji okida slot *onResultChanged* u okviru koga se ažurira kontrola za prikaz rezultata i kalkulator prikazuje 1.

- Korisnik pritiska dugme +, nakon čega dugme šalje signal *clicked* koji okida slot *onBtnClicked*. U okviru *onBtnClicked* se ispituje koje dugme je poslalo signal i nakon toga se poziva metod *doCommand* instance *CalculatorLogic* i prosleđuje "+". *CalculatorLogic* ažurira stanje rezultata i pamti da je poslednja operacija bila +.

- Korisnik pritiska dugme 2, nakon čega dugme šalje signal *clicked* koji okida slot *onBtnClicked*. U okviru *onBtnClicked* se ispituje koje dugme je poslalo signal i nakon toga se poziva metod *doCommand* instance *CalculatorLogic* i prosleđuje "2". *CalculatorLogic* ažurira stanje rezultata i šalje signal *resultChanged* koji okida slot *onResultChanged* u okviru koga se ažurira kontrola za prikaz rezultata i kalkulator prikazuje 2.

- Korisnik pritiska dugme =, nakon čega dugme šalje signal *clicked* koji okida slot *onBtnClicked*. U okviru *onBtnClicked* se ispituje koje dugme je poslalo signal i nakon toga se poziva metod *doCommand* instance *CalculatorLogic* i prosleđuje "=". *CalculatorLogic* узима у обзир претходна два броја и операцију + и на основу тога ažurira stanje rezultata i šalje signal *resultChanged* koji okida slot *onResultChanged* u okviru koga se ažurira kontrola za prikaz rezultata i kalkulator prikazuje 3. *CalculatorLogic* šalje i signal *resultHistoryChanged* koji okida slot *onResultHistoryChanged* u okviru koga se ažurira kontrola za prikaz istorije izračunavanja rezultata i prikazuje "1 + 2 = 3" u okviru *PlainTextEdit* kontrole.