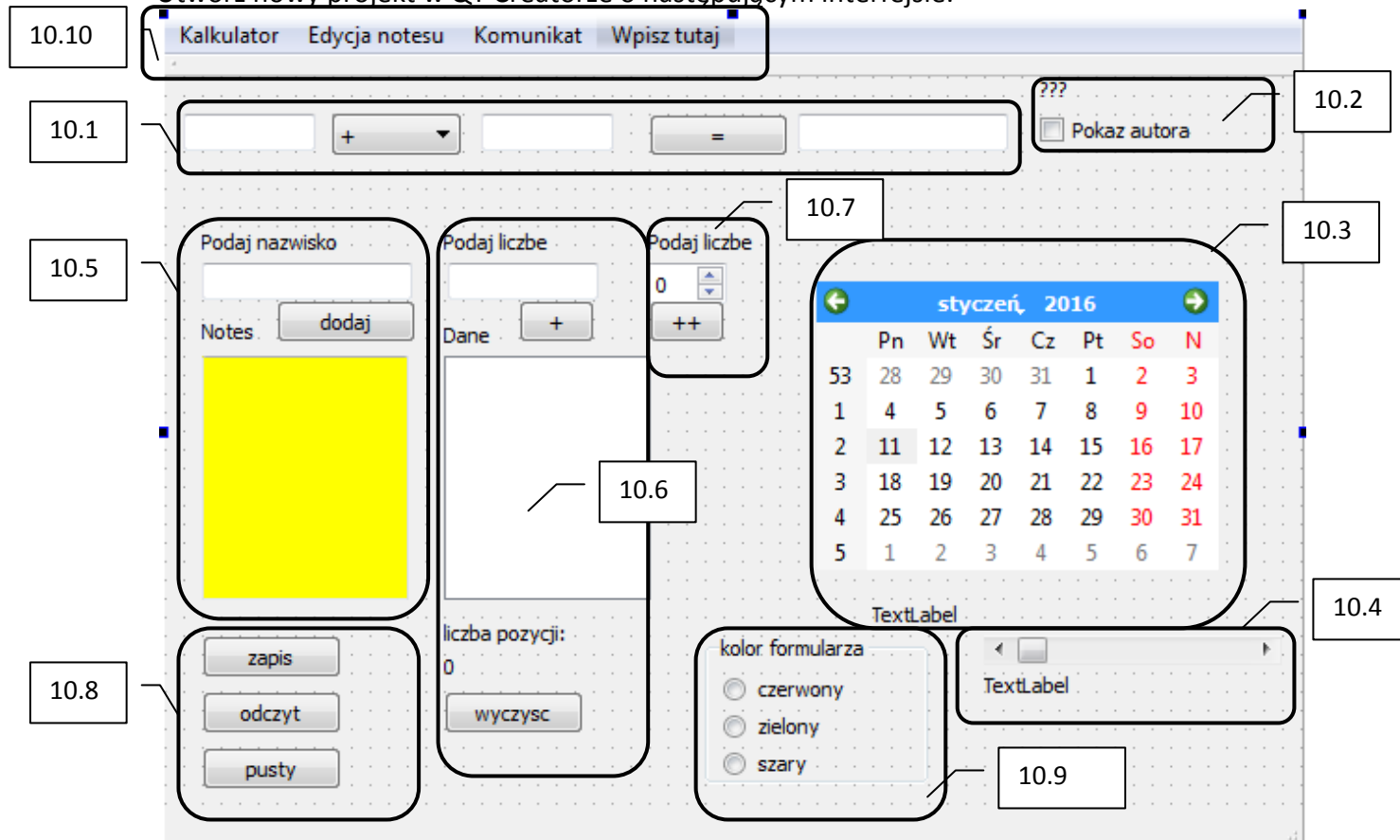


## Laboratorium 10. Wprowadzenie do biblioteki QT

### Cel laboratorium:

Zapoznanie z wykorzystaniem wybranych komponentów QT Creatora

Utwórz nowy projekt w QT Creatorze o następującym interfejsie:



### Zad.10.1. Kalkulator(*lineEdit*, *pushButton*, *comboBox*)

- Umieść wybrane komponenty do obliczeń, zmieniając ich właściwości:  
*objectName*: a, operacja, b, oblicz, wynik  
*text*: =(pushButton)
- Wprowadź pozycje pola kombi: prawy klawisz myszy *Modyfikuj elementy*
- Oprogramuj slot dla sygnału clicked() dla przycisku oblicz: prawy klawisz myszy  
Przejdź do slotu  
Wykorzystaj funkcje konwersji tekstu na liczbę  
`ui->lineEdit->text().toFloat()`  
i liczby na tekst `ui->lineEdit->setText(QString::number(liczba);`

### Zad.10.2. Autor (*checkBox*, *label*)

- Umieść wybrane komponenty, zmień ich właściwości *objectName* i *text*
- Oprogramuj slot dla sygnału clicked() dla checkBoxa, aby przy zaznaczeniu (checked) wyświetlił nazwisko autora w komponencie label

**Zad.10.3. Wprowadzanie, wyświetlanie daty (calendarWidget, label)**

- Umieść wybrane komponenty
- Oprogramuj slot dla sygnału *clicked()* dla *calendarWidget*, aby przy zaznaczeniu daty(*selectedDate()*) wyświetlił ją w komponencie *label*. Wykorzystaj metodę - funkcję konwersji daty do stringu *toString()*

**Zad.10.4. Wprowadzanie, wyświetlanie liczby (horizontalScrollBar, label)**

- Umieść wybrane komponenty
- Oprogramuj slot dla sygnału *actionTriggered()* dla *horizontalScrollBar*, aby przy przesuwaniu suwaka pozycję (*value*) wyświetlił ją w komponencie *label*. Wykorzystaj odpowiednią funkcję konwersji

**Zad.10.5. Notes - wprowadzanie, wyświetlanie tekstu (lineEdit, textEdit, label, pushButton)**

- Umieść wybrane komponenty, zmień ich właściwości *objectName*, *text*
- Dla komponentu *textEdit* zmień kolor tekstu (czerwony) i kolor tła (żółty) we właściwości *stylesheet* za pomocą pola kombi *Dodaj kolor*:  
`color: rgb(255, 0, 0);`  
`background-color: rgb(255, 255, 0);`
- Oprogramuj slot dla sygnału *clicked()* dla przycisku *dodaj*, aby tekst z komponentu *lineEdit* (nazwisko) dodał do komponentu *textEdit* (*notes*) - wykorzystaj metodę *append()*

**Zad.10.6. Sumowanie liczb - wprowadzanie, wyświetlanie liczb (lineEdit, listWidget, label, pushButton)**

- Umieść wybrane komponenty, zmień ich właściwości *objectName*, *text*
- Oprogramuj slot dla sygnału *clicked()* dla przycisku *+*, aby tekst z komponentu *lineEdit* (*dane*) dodał do komponentu *listWidget* (*lista*) - wykorzystaj metodę *addItem()* pod listą w komponentach *label* powinna wyświetlać się liczba pozycji (*count()*) i aktualna suma wprowadzonych liczb - wykorzystaj odpowiednie funkcje konwersji
- Oprogramuj slot dla sygnału *clicked()* dla przycisku *wyczyszc*, który czyści zawartość komponentu *lista* (*clear()*)

**Zad.10.7. Sumowanie liczb - wprowadzanie, wyświetlanie liczb (spinBox, listWidget, label, pushButton)**

- Umieść wybrane komponenty, zmień ich właściwości *objectName*, *text*
- Oprogramuj slot dla sygnału *clicked()* dla przycisku *++*, aby liczbę z komponentu *spinBox* (*dane2*) dodał do komponentu *listWidget* (*lista*) - wykorzystaj metodę *addItem()* i właściwość *value* dla *spinBoxa* pod listą w komponentach *label* powinna wyświetlać się liczba pozycji (*count()*) i aktualna suma wprowadzonych liczb - wykorzystaj odpowiednie funkcje konwersji

**Zad.10.8. Notes - zapis i odczyt z pliku (textEdit, pushButton)**

- Umieść wybrane komponenty, zmień ich właściwości *objectName*, *text*
- Oprogramuj slot dla sygnału *clicked()* dla przycisku *zapis*, aby tekst z komponentu *textEdit* (*notes*) został zapisany w pliku *notes.txt*.

Dodaj odpowiednie moduły:

```
#include <QFile>
```

```
#include <QString>
#include <QDebug>
#include <QTextStream>
```

przeanalizuj i wykorzystaj podany kod:

```
QString filename = "notes.txt";
QFile file(filename);
// Trying to open in WriteOnly and Text mode
if(!file.open(QFile::WriteOnly |
              QFile::Text))
{
    qDebug() << " Could not open file for writing";
    return;
}

// To write text, we use operator<<(),
// which is overloaded to take
// a QTextStream on the left
// and data types (including QString) on the right

QTextStream out(&file);
out<<ui->notes->toPlainText();

file.flush();
file.close();
```

- Oprogramuj slot dla sygnału *clicked()* dla przycisku pusty, aby tekst z komponentu *textEdit (notes)* został usunięty.
- Oprogramuj slot dla sygnału *clicked()* dla przycisku odczyt, aby zawartość pliku *notes.txt* została wyświetlona w komponencie *textEdit (notes)*

przeanalizuj i wykorzystaj podany kod:

```
QString filename = "notes.txt";
QFile file(filename);
if(!file.open(QFile::ReadOnly |
              QFile::Text))
{
    qDebug() << " Could not open the file for reading";
    return;
}

QTextStream in(&file);
QString myText = in.readAll();
ui->notes->setPlainText(myText);

/* QTextStream in(&file);
while(!in.atEnd()) {
    QString line = in.readLine();
    ui->notes->append(line);}
*/
file.close();
```

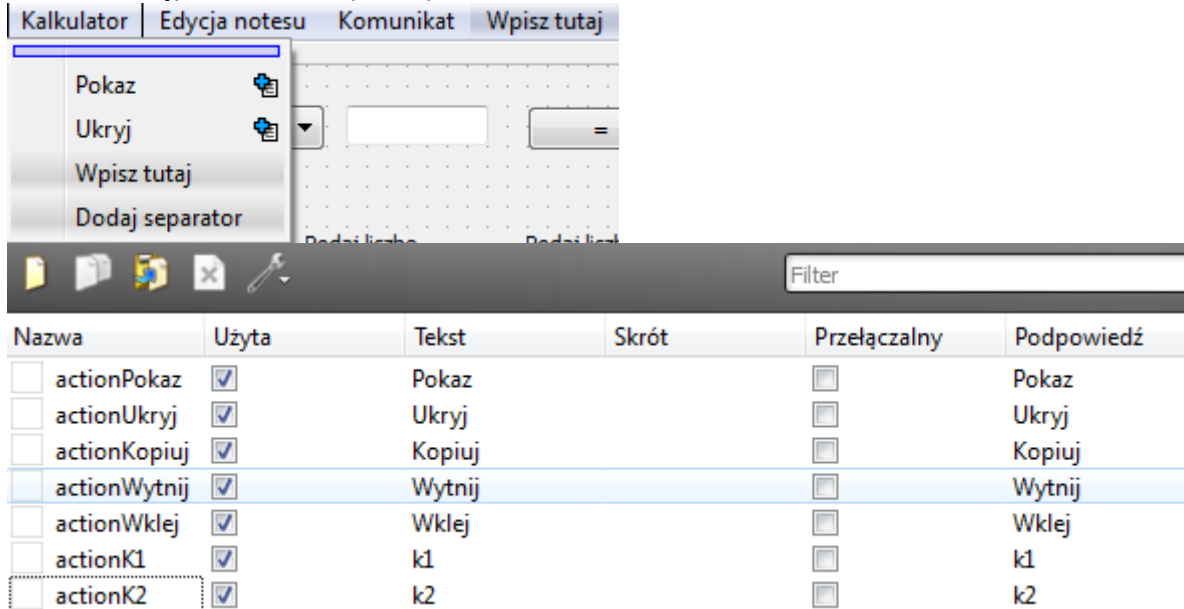
#### Zad.10.9. Kontenery(*groupBox*, *radioButton*)

- Umieść wybrane komponenty, zmień ich właściwości *objectName*, *text*
- Oprogramuj sloty dla sygnałów *clicked()* dla przycisków radiowych, aby pozwalały na zmianę koloru formularza (*MainWindow*), wykorzystaj metodę *isChecked()*

sprawdzając zaznaczenie przycisku i metodę `setStyleSheet("background-color: red")` zmieniającą kolor tła komponentu MainWindow.

#### Zad.10.10. Menu główne (menuBar)

- Wprowadź pozycje menu: Kalkulator (Pokaż, Ukryj), Edycja notesu (Kopiuj, Wytnij, Wklej), Komunikat (k1,k2)



Nazwa	Użyta	Tekst	Skrót	Przełączalny	Podpowiedź
<input type="checkbox"/> actionPokaż	<input checked="" type="checkbox"/>	Pokaż		<input type="checkbox"/>	Pokaż
<input type="checkbox"/> actionUkryj	<input checked="" type="checkbox"/>	Ukryj		<input type="checkbox"/>	Ukryj
<input type="checkbox"/> actionKopiuj	<input checked="" type="checkbox"/>	Kopiuj		<input type="checkbox"/>	Kopiuj
<input type="checkbox"/> actionWytnij	<input checked="" type="checkbox"/>	Wytnij		<input type="checkbox"/>	Wytnij
<input type="checkbox"/> actionWklej	<input checked="" type="checkbox"/>	Wklej		<input type="checkbox"/>	Wklej
<input type="checkbox"/> actionK1	<input checked="" type="checkbox"/>	k1		<input type="checkbox"/>	k1
<input type="checkbox"/> actionK2	<input checked="" type="checkbox"/>	k2		<input type="checkbox"/>	k2

- Oprogramuj sloty dla akcji Pokaż/Ukryj (sygnał `triggered()`) wykorzystując metodę `setVisible()` z argumentem typu logicznego, która pokaże lub ukryje wszystkie komponenty dla kalkulatora  
Napisz instrukcje, aby w momencie uruchomienia programu komponenty dla kalkulatora były niewidoczne
- Oprogramuj sloty dla akcji Kopiuj, Wytnij, Wklej pozwalające wykonywać odpowiednie operacje blokowe na tekście komponentu notes.
- Oprogramuj sloty dla akcji k1, k2 demonstrujące przykładowe komunikaty wyświetlane w oknach dialogowych:
 

```
QMessageBox::information(this, "Komunikat 1", "informacja");
QMessageBox::warning(this, "Komunikat 1", "ostrzezenie",
                    QMessageBox::Yes | QMessageBox::No | QMessageBox::Cancel,
                    QMessageBox::Yes);
```
- Dodaj komunikaty innych typów.