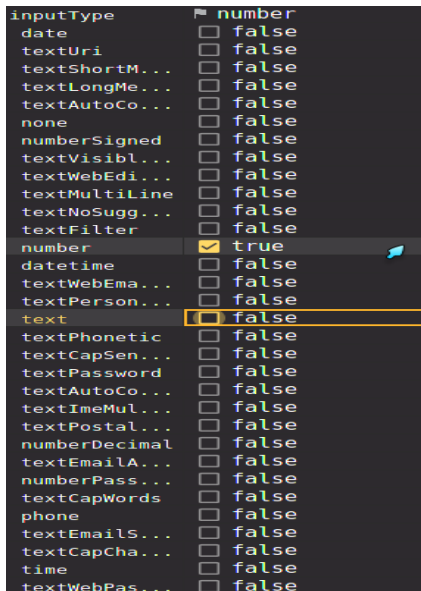


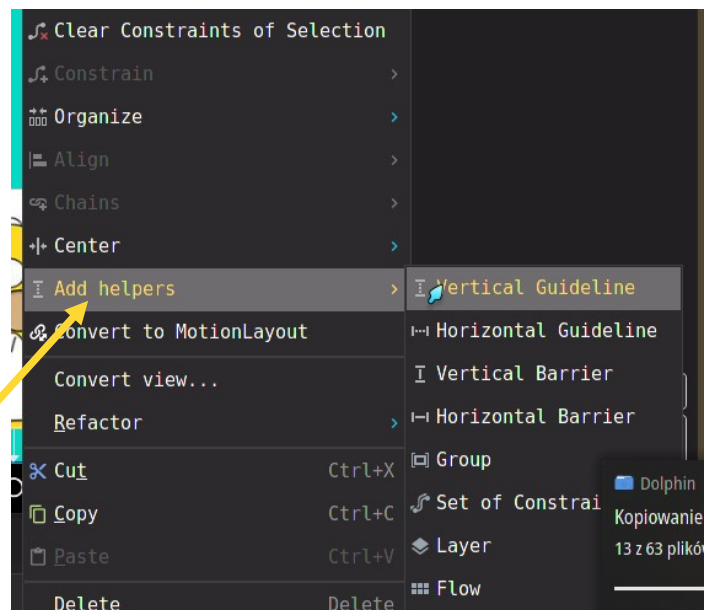
# 3 zadania

Stwórz kalkulator w android studio.

Aby do TextView można było wprowadzać tylko odpowiednie wartości (liczby, tekst itp.) należy kliknąć: **inputType**



Następnie wybrać z dostępnych opcji co chcemy aby można było wprowadzać w danym elemencie!



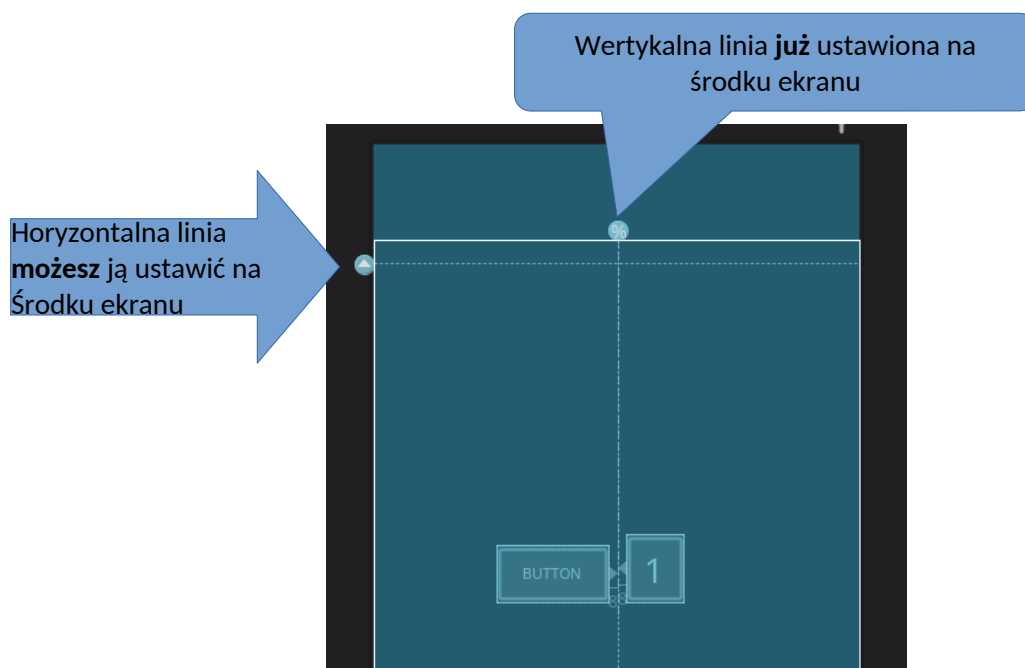
Aby elementy znajdowały się idealnie na środku ekranu należy:

kliknąć pp myszy na Layout i wybrać

Zobacz co pojawi się na ekranie:

(Blueprint)

Będzie to linia, którą można ustawić idealnie na środku ekranu i do niej przymocować przyciski naszego kalkulatora, pomaga ustawić elementy UI idealnie na środku ekranu – niezależnie od rozdzielczości.



# 3 zadania

Aby aplikacja reagowała na wciśnięcie przycisku należy użyć następującego fragmentu kodu (Czyli na przycisku **addOneButton** używamy metody **setOnClickListener**):

```
addOneButton.setOnClickListener(view -> {  
    //Tutaj kod programu który chcemy wykonać po naciśnięciu przycisku  
    number++; //zwiększenie wartości o 1 - INKREMENTACJA  
    tvNumber.setText(""+number); // Przypisanie nowego tekstu do widoku TextView  
});
```

Zwróć uwagę na nawiasy i klamierki!!!!

## UWAGA:

Stwórz zmienną globalną do przechowywania danych (Tworzysz nową klasę), z tych zmiennych będziesz wstawiał dane do widoku wstawiał do widoku! Jak pokazano wyżej. Czyli nie używamy do elementów widoku metody **getText()**!!! Pamiętaj że do danych dostajemy się za pomocą **set** i **get**.

```
java  
└─ example.com.myapplication  
    └─ CalculatorData  
    └─ MainActivity  
3 public class CalculatorData {  
4     //tutaj wszystkie dane jakie będzie przechowywał nasz kalkulator  
5     //Pomyśl jak można stworzyć historię wykonywanych działań  
6     private int number = 0;  
7 }
```

Tworzenie getterów i setterów za pomocą IDE:

```
public class CalculatorData {  
    //tutaj wszystkie dane jakie będzie przechowywał nasz kalkulator  
    //Pomyśl jak można stworzyć historię wykonywanych działań  
    private int number = 0;  
}
```

PP Myszy i wybieramy jak niżej (skrót klawiaturowy masz obok Generate ...)

Show Context Actions Alt+Enter

Paste Ctrl+V

Copy / Paste Special >

Column Selection Mode Alt+Shift+Insert

Refactor >

Folding >

Analyze >

Go To >

Generate... Alt+Insert

Open In >

Local History >

Compare with Clipboard

Create Gist...

Pojawi się okno z wyborami:

Generate

Constructor

Getter

Setter

Getter and Setter

equals() and hashCode()

toString()

Override Methods... Ctrl+O

Test...

Copyright

Kotlin data classes from JSON Alt+K

Wybierz co chcesz wygenerować i zobacz co się stanie!!!  
Pamiętaj zawsze możesz cofnąć zamiany!