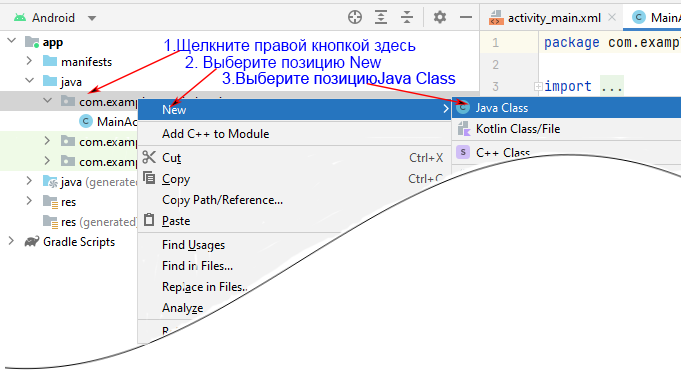
Элементы гипертекста

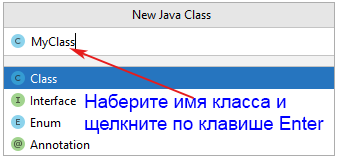
Переход на гипертекст Ctrl+мышь, возврат из гипертекста Alt+ ←.

Создание класса (p1)

1. Вызов окна для набора имени класса



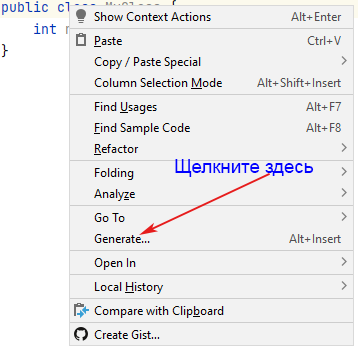
1. Окно набора имени класса



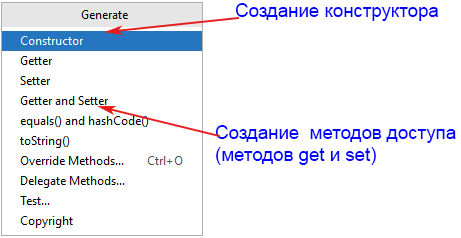
Имя класса всегда набирают с заглавной буквы.

Формирование элементов класса (p2)

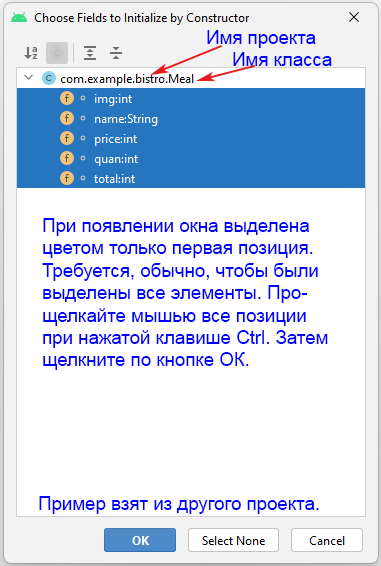
Щелкните правой кнопкой по строке заголовка класса. Появится окно:



В этом окне щелкните по строке Generate. Появится окно



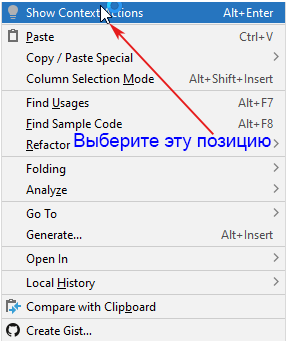
После этого появится окна для выбора элементов класса:



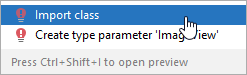
В появившемся окне отметьте все позиции. Если позиция будет пропущена, для не будет выполнено действие (пропущен параметр конструктора или пара методов get and set).

Импорт класса (p3)

1. Щелкните правой кнопкой по имени класса, подчеркнутому красной линией, и в появившемся окне выберите позицию Show Context Actions.

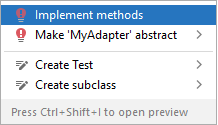


1. В появившемся затем окне щелкните по позиции Import class



Добавление заготовок для требуемых методов (p4)

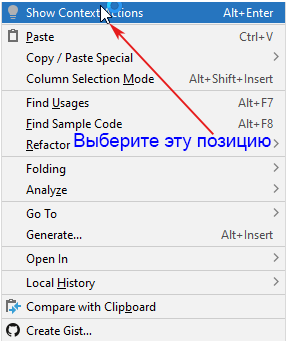
Щелкните правой кнопкой по строке заголовка адаптера (подчеркнут красной линией)



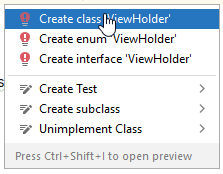
Щелкните по строке Implements methods.

Добавление класса ViewHolder (p5)

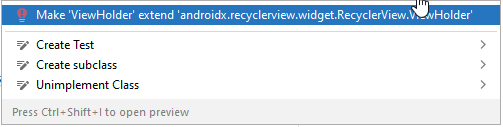
1. В заголовке адаптера щелкните правой кнопкой по строке ViewHolder, появится окно:



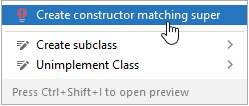
1. Появится окно



1. Щелкните по выделенной на скриншоте позиции и попадете в окно добавления наследования, в котором также щелкните по выделенной на скриншоте позиции.



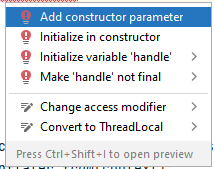
1. Теперь заголовок класса ViewHolder подчеркнут красной линией, щелкните правой кнопкой по ней. Появится окно с требованием создания конструктора для этого класса. Щелкните по выделенной на скриншоте позиции.



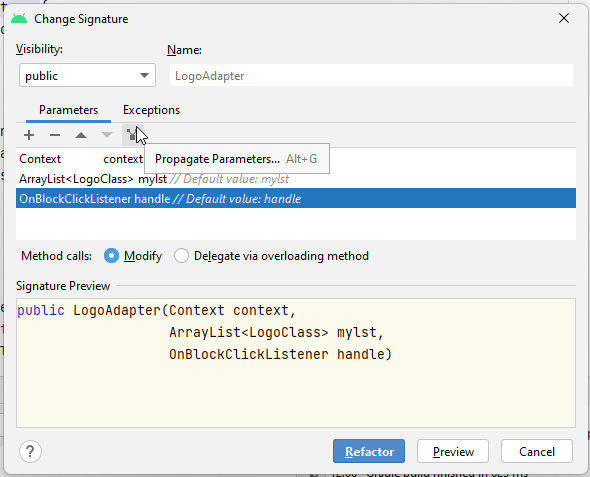
Класс ViewHolder создан полностью.

Добавление параметра в конструктор адаптера. (p6)

После добавления объявлений в классе адаптере (и других классах) добавленная строка выделяется красным цветом. Щелкните правой кнопкой по строке меню Show Context Actions и в появившемся новом меню щелкните по позиции "Add constructor parameter".



Появится окно (скриншот переделать, не тот конструктор)



Щелкните по кнопке Refactor (проверьте, чтобы в окне был ваш класс)

Глобальное объявление переменных (p7)

Переменные, объявленные сразу после заголовка класса, являются глобальными (доступны во всех методах класса). Переменные, объявленные внутри любого метода, являются локальными (доступны только в этом методе).

public class MainActivity extends AppCompatActivity {

Button bt1, bt2;

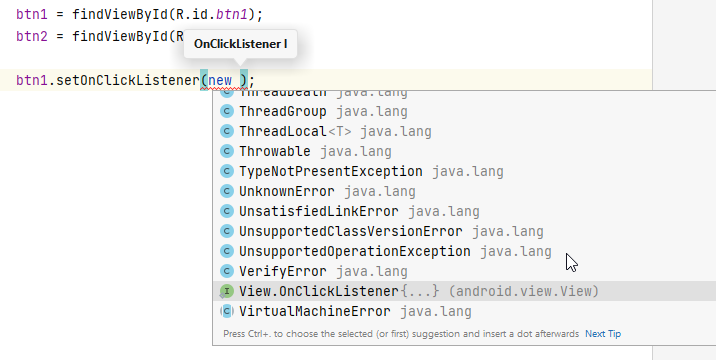
Кнопки bt1и bt2объявлены глобально (после заголовка класса MainActivity).

Добавление одиночного обработчика щелчка (h1)

После закрепления элемента за именем (findViewById) вставьте точку, а за ней одну или несколько первых букв обработчика (setO).



В появившемся списке выберите позицию setOnClick Listener



Будет добавлен обработчик щелчка по элементу

btn1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

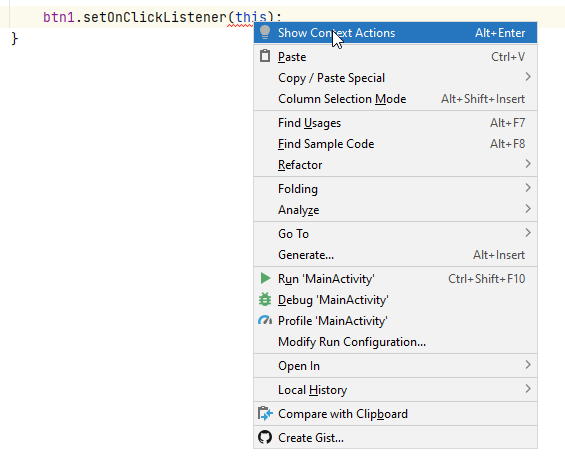
*// Действия, задаваемые данным обработчиком*

}

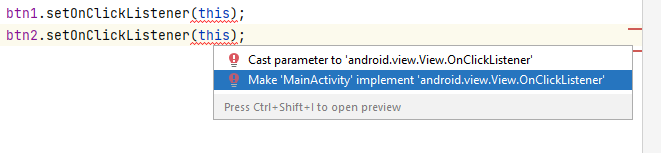
});

Данный способ следует применять в случаях, при которых количество обработчиков небольшое, поскольку они размещаются внутри метода onCreate соответствующей активнсти (окна) и теряется стройность программы.

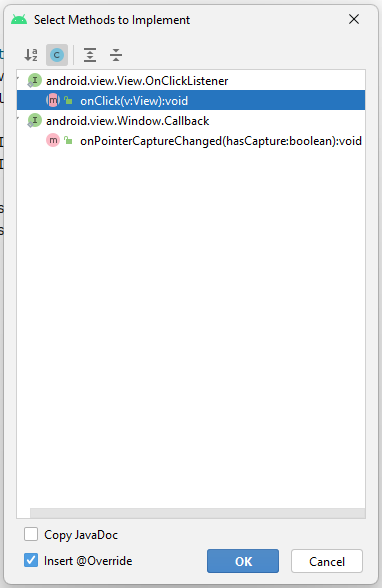
Если обработчиков несколько, то в круглых скобках вместо слова new следует набрать слово this. Эту строку надо повторить для всех элементов (одного типа), для которых добавляются обработчики щелчков (h2). Строка будет подчеркнута красным цветом. Щелкните по ней правой кнопкой. Появится меню в котором щелкните по строке Show Context Actions



Появится меню, в котором выберите вторую строку



В новом окне щелкните по кнопке ОК



Появится метод для обработки щелчков. Поскольку он применяется для нескольких щелчков, то надо в нем оформить оператор switch? Которой имеет следующую форму

@Override

public void onClick(View v) {

}

Обработка нескольких щелчков h2

@Override

public void onClick(View v) {

switch (v.getId()){

case R.id.*btn1*:

*// Действия, выполняемые при щелчке по кнопке*

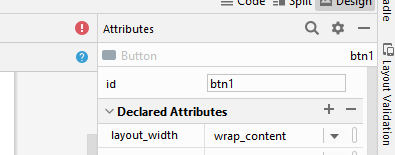
case R.id.*btn2*:

*// Действия, выполняемые при щелчке по кнопке*

break;

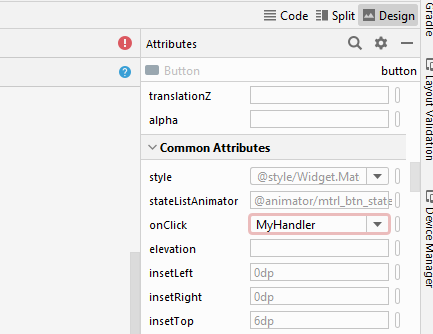
}

}



Обработка множества щелчков (h3)

При большом количестве обработчиков щелчков объявляют метод (имя произвольное) для обработки, внутри которого обработка для каждого элемента реализована с помощью оператора switch (см. обработку по закладке h2). В файл разметки вставляют оператор, содержащий имя метода для обработки, например: android:onClick="MyHandler" (MyHandler имя метода). После этого в режиме Design обработки файла разметки в позиции OnClick указывают имя созданного метода (Эта позиция находится в группе Common Attributes)



Закрепление элементов за разметкой (p8)

Метод класса MainActivity :

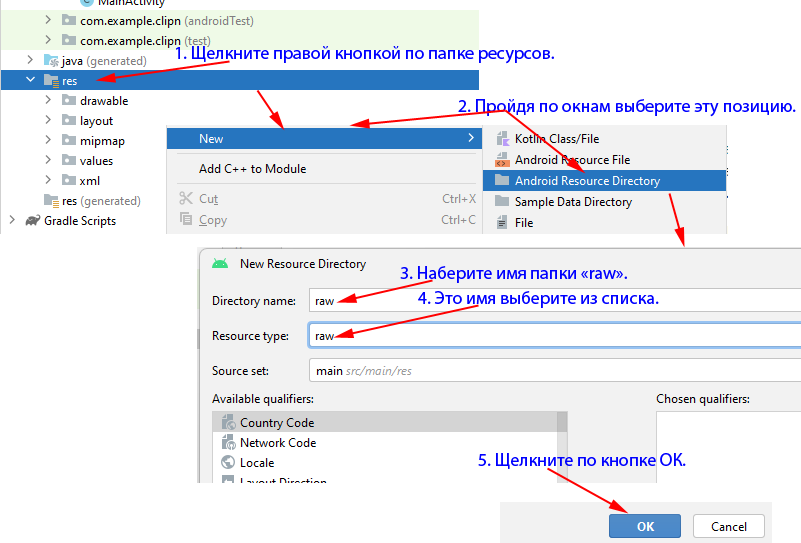
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
*// В этом месте вставляют операторы, например, закрепления*

txtA= findViewById(R.id.*txtA*);

Добавление элементов в разметку (p9)

Новая закладка (p10)

Создание папки ресурсов (для звуковых и видеофайлов)



Список закладок

Создание класса (p1)

Формирование элементов класса (p2)

Импорт класса (p3)

Добавление заготовок для требуемых методов (p4)

Добавление класса ViewHolder (p5)

Добавление параметра в конструктор адаптера. (p6)

Глобальное объявление переменных (p7)

Закрепление элементов за разметкой (p8)

Добавление элементов в разметку (p9)

Добавление одиночного обработчика щелчка (h1)

Обработка нескольких щелчков h2

Обработка множества щелчков (h3)

Создание папки raw (р10)