



*Белорусский государственный университет  
информатики и радиоэлектроники*

---

## **Эргономика мобильных приложений**

**Преподаватель: Меженная Марина Михайловна**

К.Т.Н., доцент,

доцент кафедры инженерной психологии и эргономики  
а 609-2

[mezhennaya@bsuir.by](mailto:mezhennaya@bsuir.by)



---

*Кафедра инженерной психологии и эргономики*

# Лекция 11: Использование фоновых потоков

---

## План лекции:

1. `AsyncTask`: простой пример.
2. `AsyncTask`: сложный пример.

# Класс AsyncTask

---

Класс AsyncTask предлагает простой и удобный механизм для перемещения трудоёмких операций в фоновый поток.

AsyncTask создает, синхронизирует потоки, а также управляет ими, что позволяет создавать асинхронные задачи, состоящие из операций, выполняющихся в фоновом режиме, и обновлять пользовательский интерфейс по их завершении.

# 1. Создаем класс-наследник от AsyncTask

---

```
public class MyTask extends AsyncTask <[Input Parameter Type],  
[Progress Report Type], [Result Type]>
```

1. Тип входных данных

2. Тип промежуточных данных, которые используются для вывода промежуточных результатов

3. Тип возвращаемых данных

Если не нужно принимать параметры, обновлять информацию о ходе выполнения или выводить конечный результат, просто укажите тип Void во всех трёх случаях:

```
public class MyTask extends AsyncTask<Void, Void, Void>
```

---

## 2. Переопределяем основные методы AsyncTask

---

**doInBackground()** — основной метод, который выполняется в новом потоке. Не имеет доступа к UI. Именно в этом методе должен находиться код для тяжелых задач. Принимает набор параметров тех типов, которые определены в реализации вашего класса. Этот метод выполняется в фоновом потоке, поэтому в нем не должно быть никакого взаимодействия с элементами пользовательского интерфейса.

**onPostExecute()** — выполняется перед **doInBackground()**. Имеет доступ к UI.

**onPostExecute()** — выполняется после **doInBackground()** (может не вызываться, если AsyncTask был отменен). Имеет доступ к UI. Используйте его для обновления пользовательского интерфейса, как только ваша фоновая задача завершена.

---

## 2. Переопределяем основные методы AsyncTask

---

**onProgressUpdate.** Имеет доступ к UI. Переопределите этот обработчик для публикации промежуточных обновлений в пользовательский интерфейс. При вызове он синхронизируется с потоком GUI, поэтому в нём вы можете безопасно изменять элементы пользовательского интерфейса.

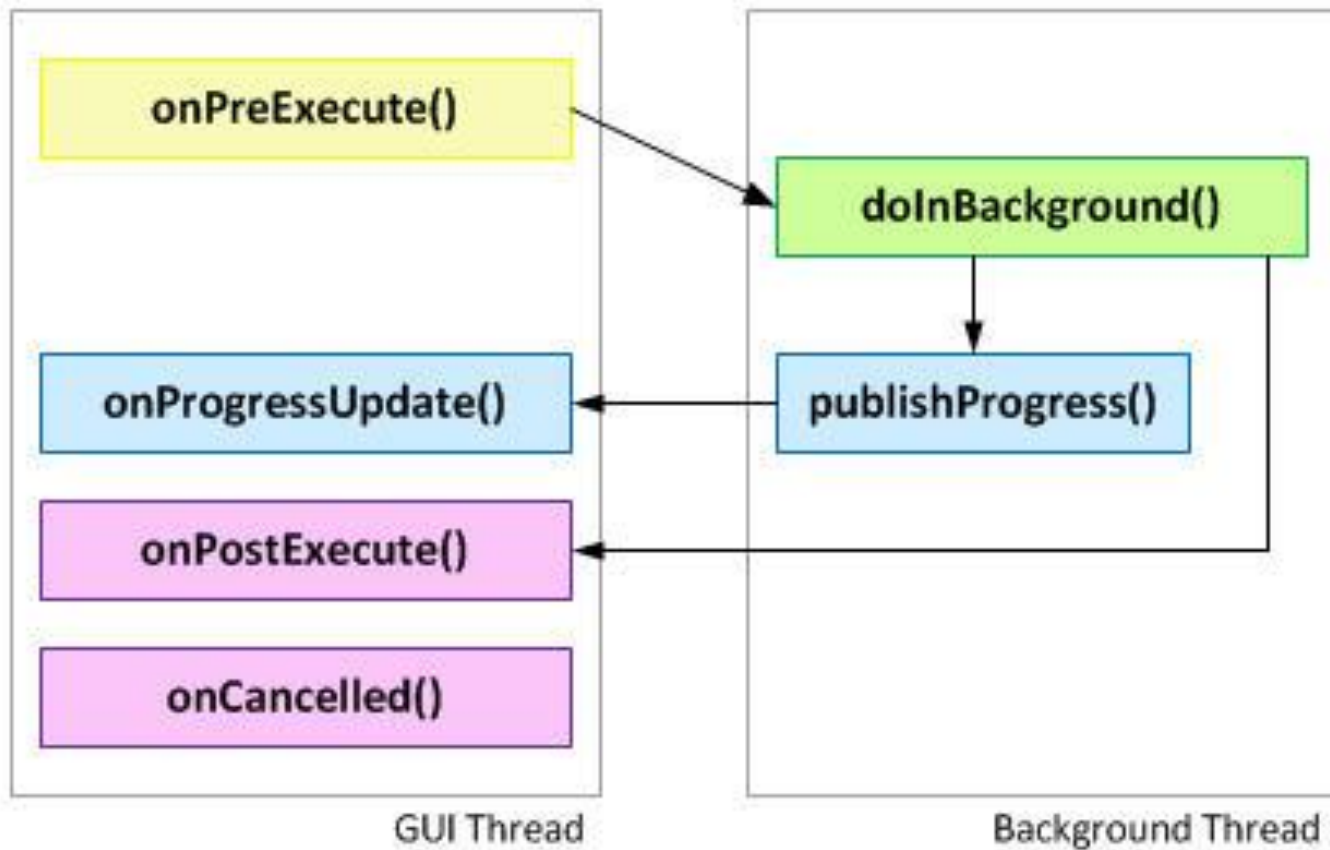
Для активации `onProgressUpdate` в методе `doInBackground()` необходимо вызвать метод **`publishProgress()`**.

**`onCancelled()`.** Имеет доступ к UI. Выполняется при отмене пользователем выполнения фонового процесса (в этом случае `onPostExecute()` не выполняется).

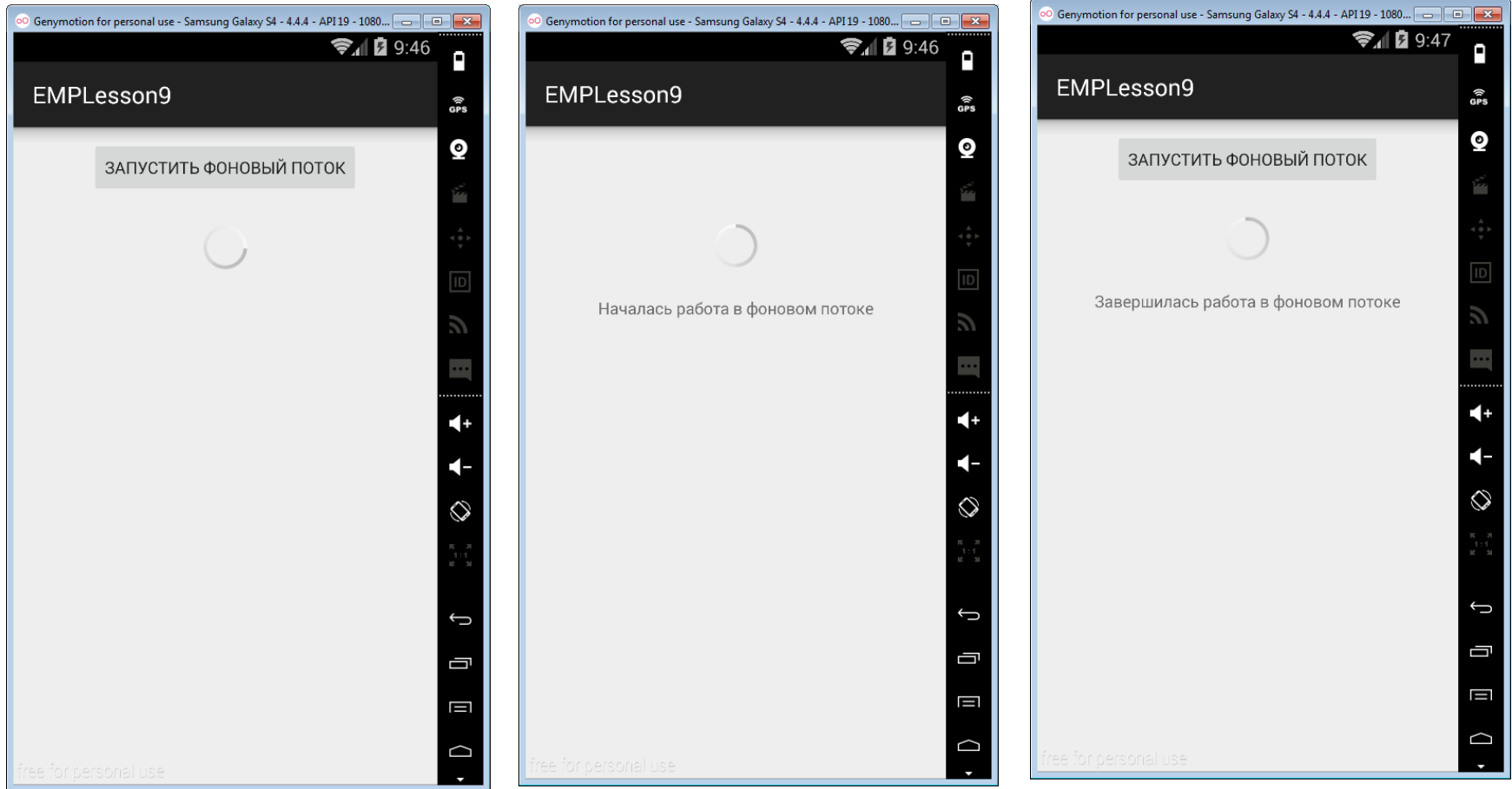
---

## 2. Переопределяем основные методы AsyncTask

---

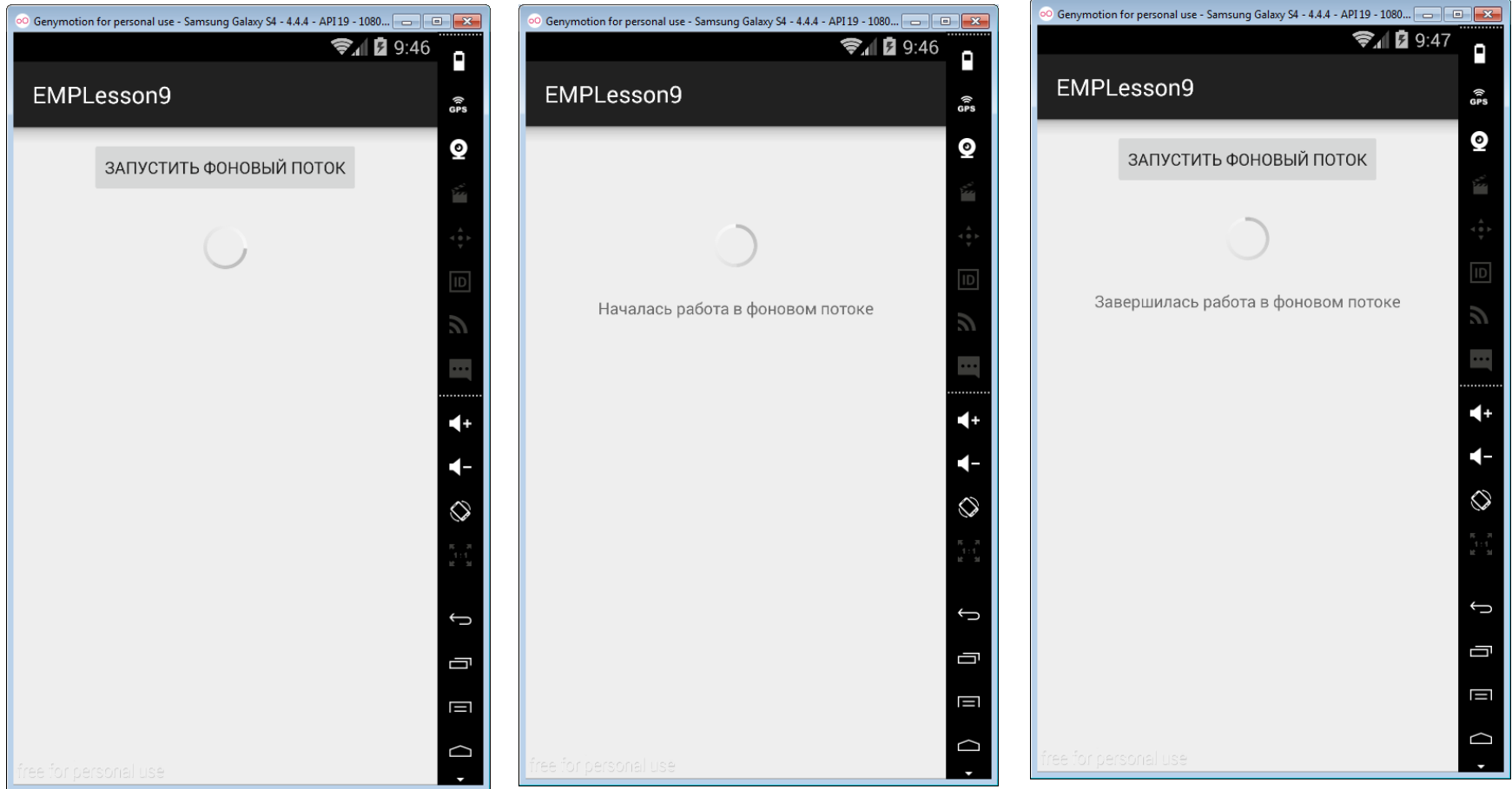


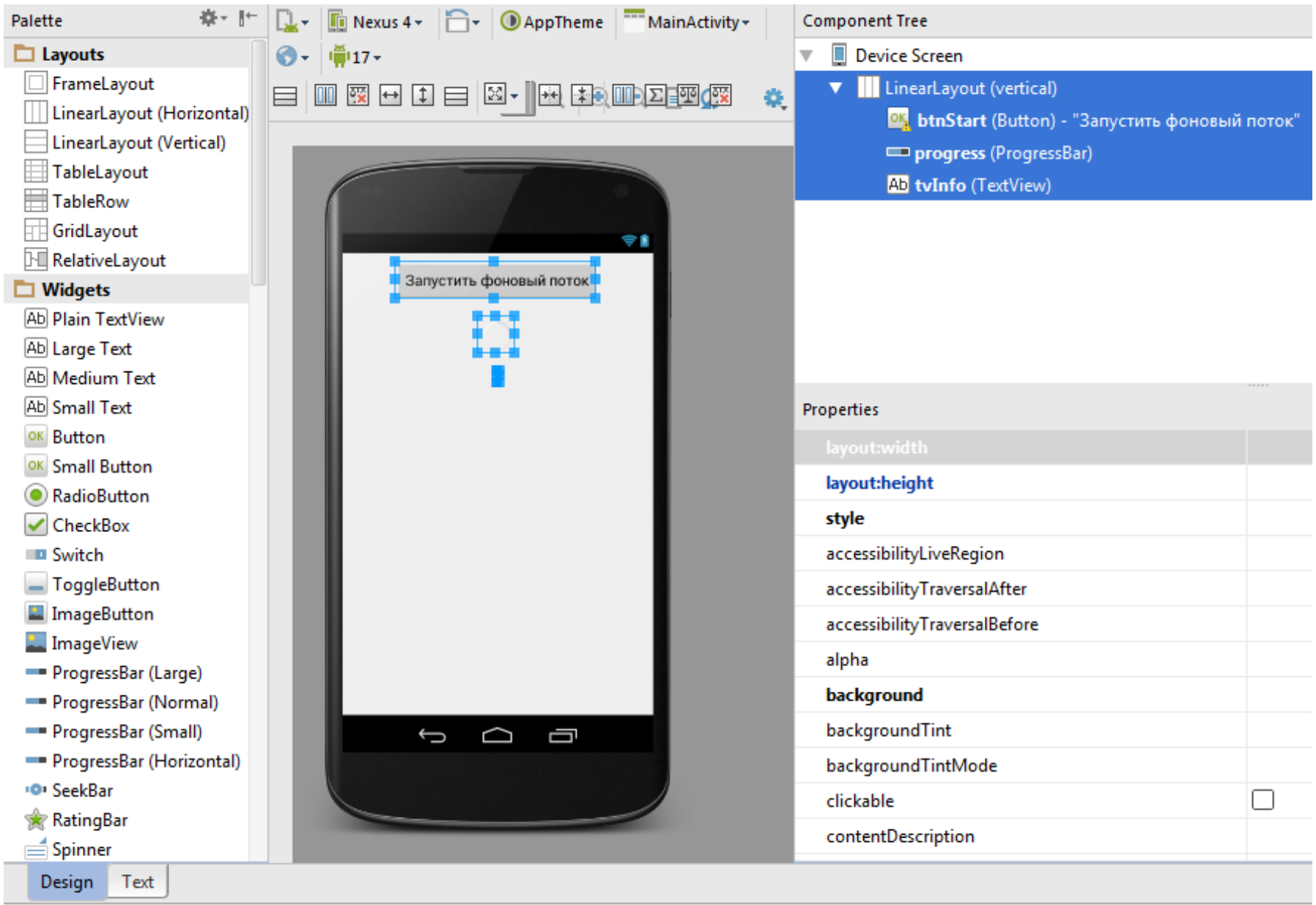
# Простой пример





# Простой пример





# Простой пример

---

```
public class MainActivity extends ActionBarActivity implements
View.OnClickListener{

    private MyTask myTask;
    private Button btnStart;
    private TextView tvInfo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }
}
```

# Простой пример

---

```
private void initView() {  
    tvInfo = (TextView) findViewById(R.id.tvInfo);  
    btnStart = (Button) findViewById(R.id.btnStart);  
    btnStart.setOnClickListener(this);  
}
```

@Override

```
public void onClick(View view) {  
    myTask = new MyTask();  
    myTask.execute();  
}
```

# Простой пример

---

**//Внутренний класс MainActivity !!!**

```
public class MyTask extends AsyncTask<Void, Void, Void> {  
  
    @Override  
    protected void onPreExecute() {  
        super.onPreExecute();  
        tvInfo.setText("Началась работа в фоновом потоке");  
        btnStart.setVisibility(View.INVISIBLE);  
    }  
}
```

# Простой пример

---

@Override

```
protected Void doInBackground(Void... voids) {  
    try {  
        TimeUnit.SECONDS.sleep(5);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    return null;  
}
```

@Override

```
protected void onProgressUpdate(Void... values) {  
    super.onProgressUpdate(values);  
}
```

# Простой пример

---

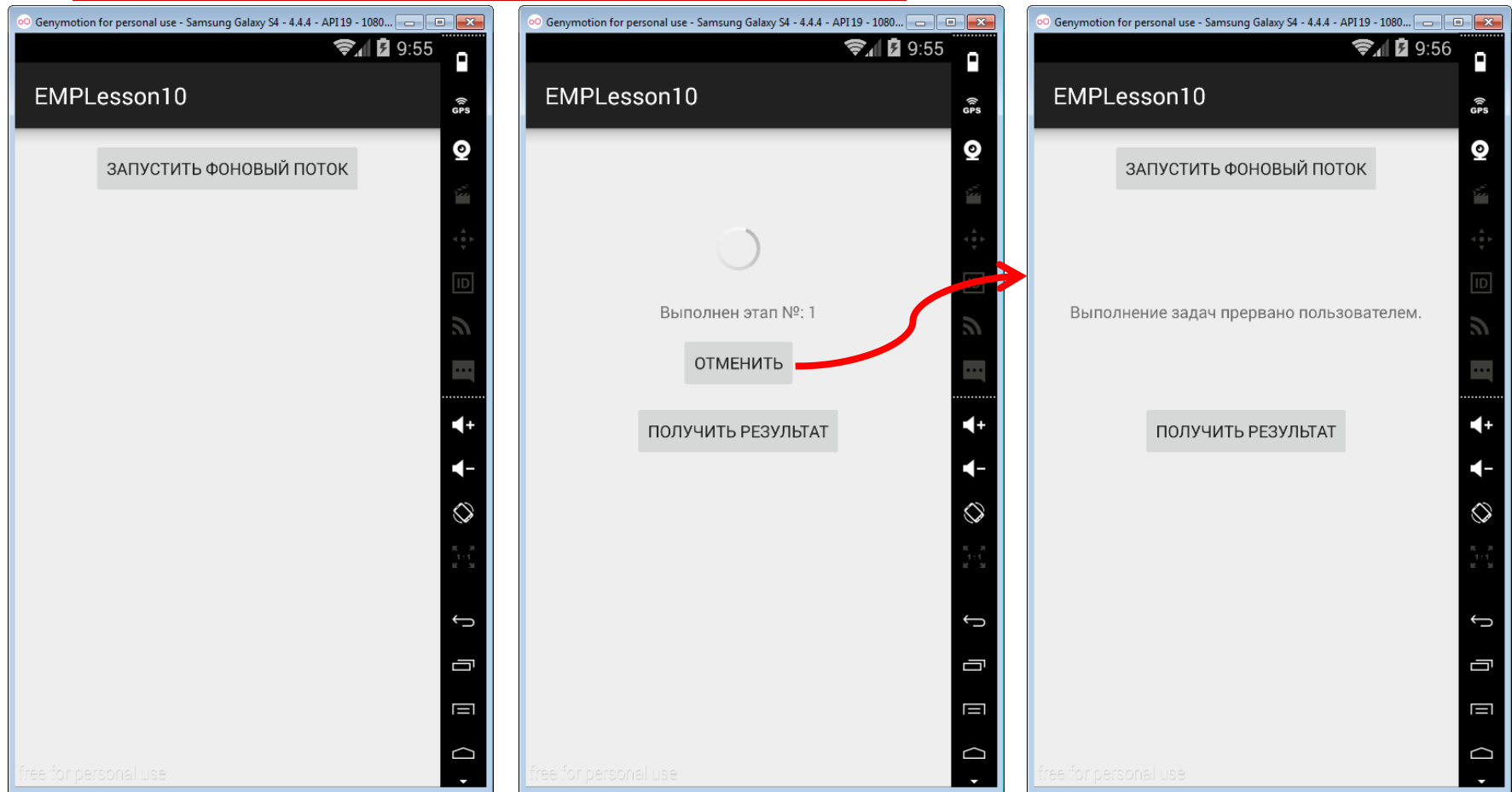
@Override

```
protected void onPostExecute(Void aVoid) {  
    super.onPostExecute(aVoid);  
    tvInfo.setText("Завершилась работа в фоновом потоке");  
    btnStart.setVisibility(View.VISIBLE);  
}
```

```
}
```

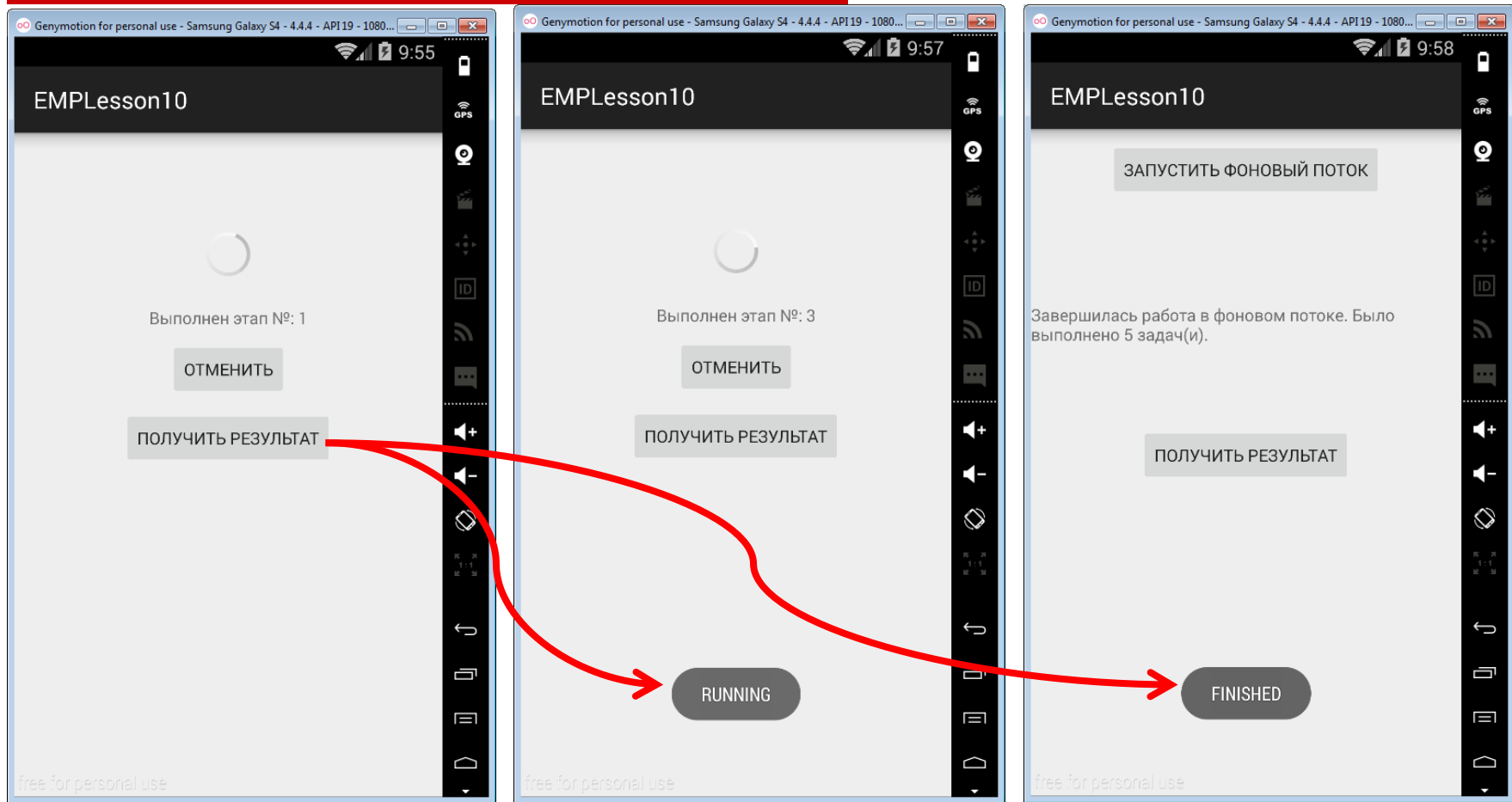
```
}
```

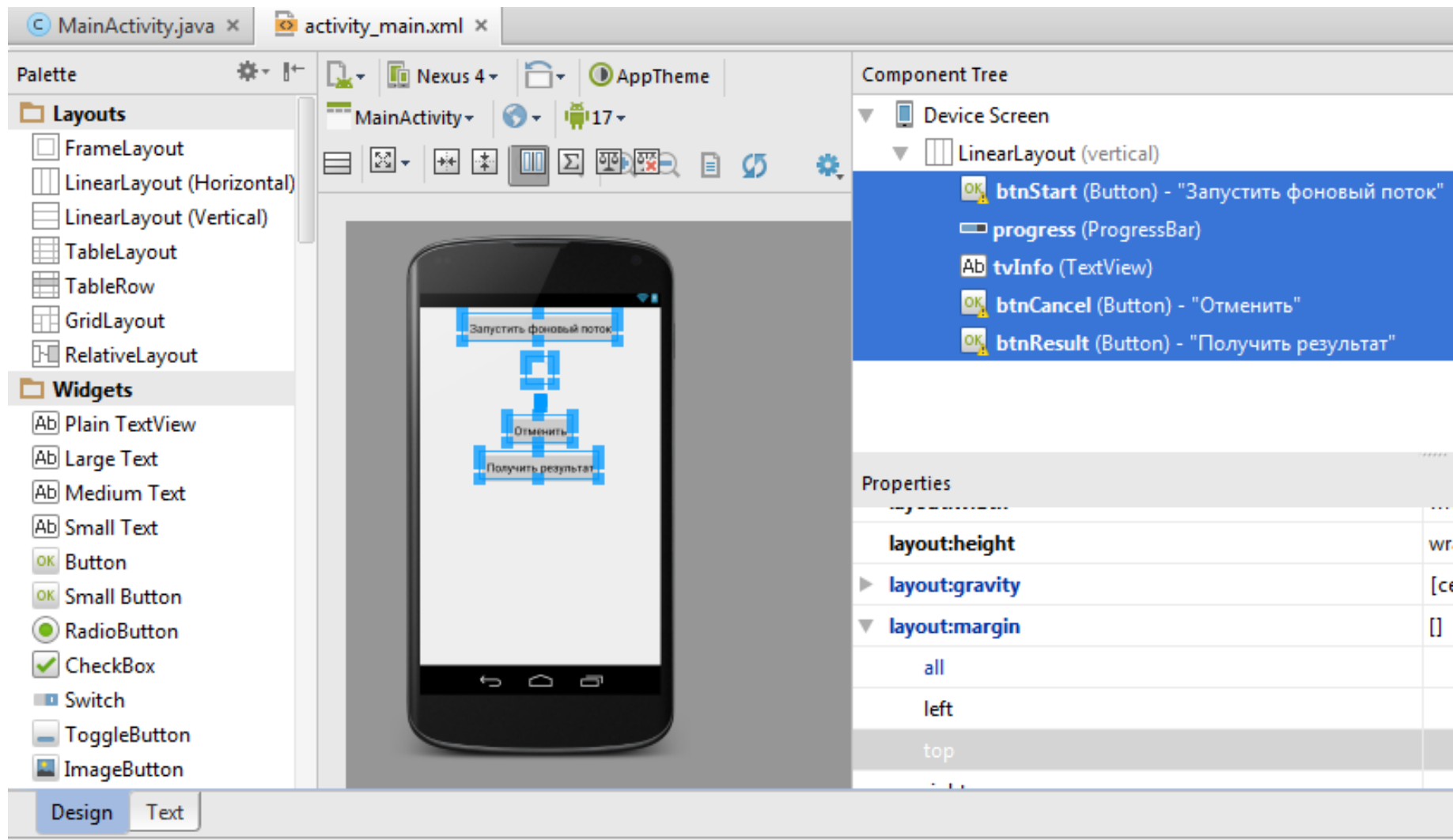
# Сложный пример





# Сложный пример





## Сложный пример

---

```
public class MainActivity extends ActionBarActivity implements
View.OnClickListener{

    private MyTask myTask;
    private Button btnStart, btnResult, btnCancel;
    private TextView tvInfo;
    private ProgressBar progress;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }
}
```

---

## Сложный пример

---

```
private void initView() {  
    tvInfo = (TextView) findViewById(R.id.tvInfo);  
    btnStart = (Button) findViewById(R.id.btnStart);  
    btnStart.setOnClickListener(this);  
    btnResult = (Button) findViewById(R.id.btnResult);  
    btnResult.setOnClickListener(this);  
    btnCancel = (Button) findViewById(R.id.btnCancel);  
    btnCancel.setOnClickListener(this);  
    progress = (ProgressBar) findViewById(R.id.progress);  
  
    btnResult.setVisibility(View.INVISIBLE);  
    btnCancel.setVisibility(View.INVISIBLE);  
    progress.setVisibility(View.INVISIBLE);  
}
```

## Сложный пример

---

@Override

```
public void onClick(View view) {  
    switch (view.getId()) {  
        case R.id.btnStart:  
            myTask = new MyTask();  
            myTask.execute("something1", "something2", "something3",  
"something4", "something5");  
            break;  
        case R.id.btnCancel:  
            myTask.cancel(true);  
            break;  
    }  
}
```

## Сложный пример

---

```
case R.id.btnResult:
    if (myTask == null) return;
    Toast.makeText(this, myTask.getStatus().toString(),
Toast.LENGTH_SHORT).show();
    try {
        int result = myTask.get(1, TimeUnit.SECONDS);
        Toast.makeText(this, "Всего выполнено " + result + "
задач(и).", Toast.LENGTH_LONG).show();
    } catch (Exception e) {
        e.printStackTrace();
    }
    break;
}
```

## Сложный пример

---

```
public class MyTask extends AsyncTask<String, Integer, Integer> {  
  
    @Override  
    protected void onPreExecute() {  
        super.onPreExecute();  
        tvInfo.setText("Началась работа в фоновом потоке");  
        btnStart.setVisibility(View.INVISIBLE);  
        progress.setVisibility(View.VISIBLE);  
        btnCancel.setVisibility(View.VISIBLE);  
        btnResult.setVisibility(View.VISIBLE);  
    }  
}
```

## Сложный пример

---

@Override

```
protected Integer doInBackground(String... my_data) {  
    try {  
        int counter = 0;  
        for (String i : my_data) {  
            doSomething(i);  
            publishProgress(++counter);  
            if (isCancelled())  
                return null;  
        }  
        TimeUnit.SECONDS.sleep(1); // разъединяемся  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    int result = my_data.length;  
    return result;  
}
```

---



## Сложный пример

---

```
private void doSomething(String i) throws InterruptedException {  
    TimeUnit.SECONDS.sleep(1);  
}
```

@Override

```
protected void onProgressUpdate(Integer... values) {  
    super.onProgressUpdate(values);  
    tvInfo.setText("Выполнен этап №: " + values[0]);  
}
```

## Сложный пример

---

@Override

```
protected void onPostExecute(Integer result) {  
    super.onPostExecute(result);  
    tvInfo.setText("Завершилась работа в фоновом потоке. Было  
выполнено " + result + " задач(и).");  
    btnStart.setVisibility(View.VISIBLE);  
    progress.setVisibility(View.INVISIBLE);  
    btnCancel.setVisibility(View.INVISIBLE);  
}
```

## Сложный пример

---

@Override

```
protected void onCancelled() {  
    super.onCancelled();  
    tvInfo.setText("Выполнение задач прервано пользователем.");  
    btnStart.setVisibility(View.VISIBLE);  
    progress.setVisibility(View.INVISIBLE);  
    btnCancel.setVisibility(View.INVISIBLE);  
}  
}
```

```
}
```