МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению лабораторных работ

по дисциплине «Программирование мобильных устройств»

для студентов направлений 09.03.03 «Прикладная информатика» и 09.03.02 «Информационные системы и технологии».

Ставрополь

2021

# **Лабораторная работа 7. Локализация и списки.**

**Цель работы:** научиться использовать ресурсы и менять содержимое в зависимости от локализации телефона.

**Формируемые компетенции:** ПК-7, ПК-8

**Теоретическая часть**

Стандартное бизнес-приложение может быть использовано не только в России. Поэтому, чтобы зарубежные пользователи могли также беспрепятственно работать с Вашим приложением, можно добавить локализацию. На первой лабораторной работе мы нашли файл со строковыми ресурсами, куда выносятся общие для приложения строки. Располагается он папке values:

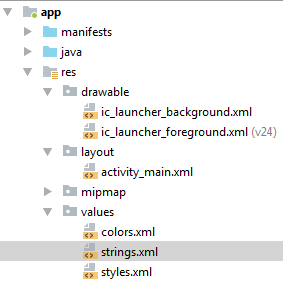


Рисунок 1 –Расположение папки values

Здесь мы можем добавлять общие значения для определённых типов. Например, в strings.xml хранится список общих строк для всего приложения. Как Вы могли заметить, если на кнопке или TextView жёстко прописать содержимое строки, то она будет выделяться коричневым цветом. Это не считается ошибкой, но является предупреждением. Лучше выводить все значения свойства android:text в файл strings.xml, так же, как и цвета в файл colors.xml.

В первую очередь необходимо разграничить ресурсы для различных стран. Поэтому, для каждой страны создадим соответствующую папку values с постфиксом краткого наименования. Так, для России создадим values-ru.

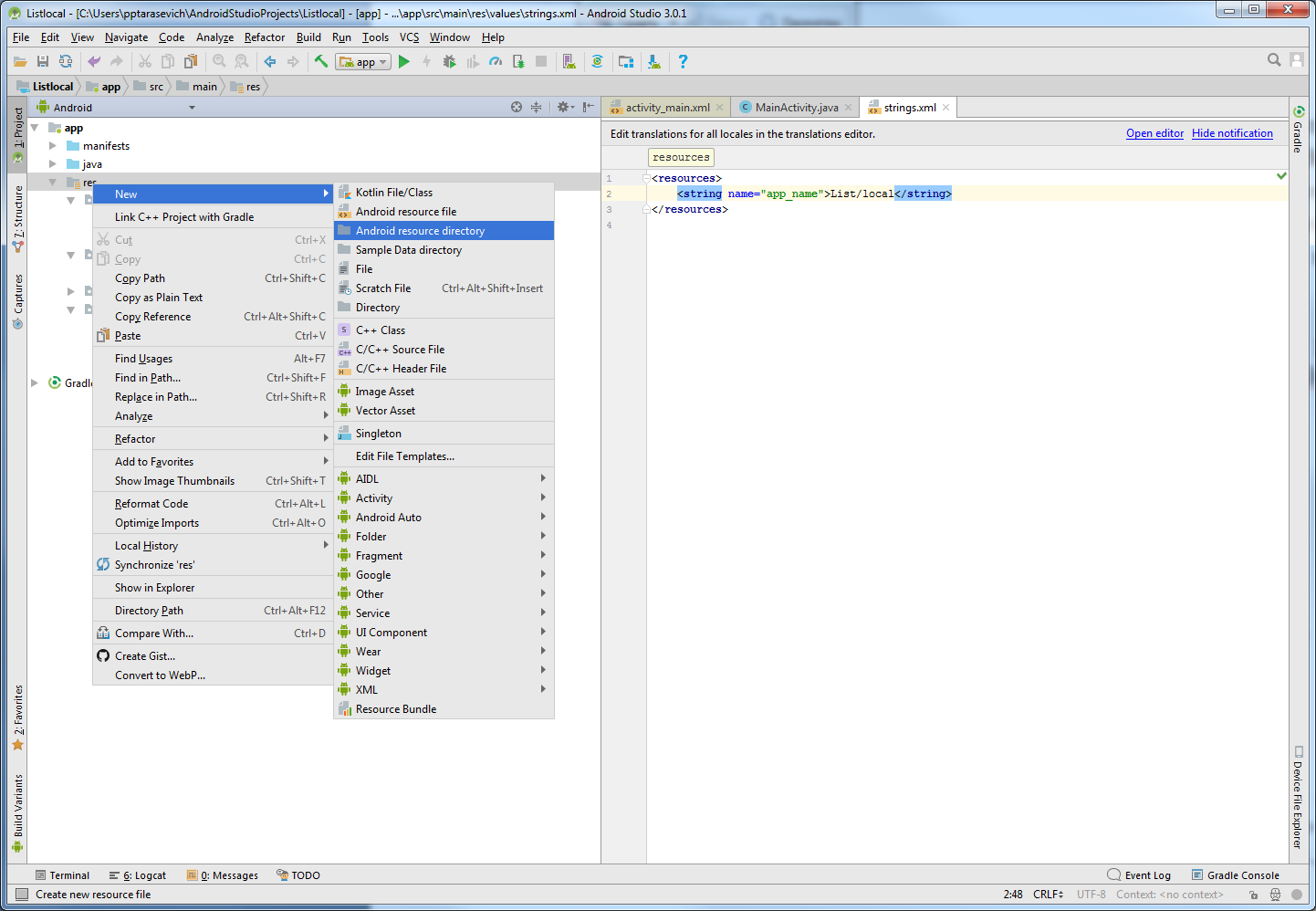


Рисунок 2 – Добавление директории

В открывшемся меню, в списке ограничений, выберем пункт Locale и нажмём кнопку, выделенную на рисунке ниже.

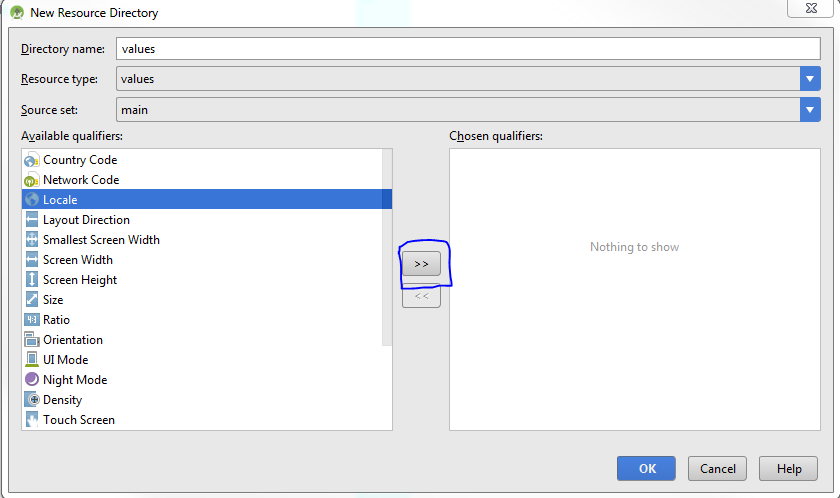


Рисунок 3 – Добавление локализаци

Найдём в списке Россию и нажмём OK.

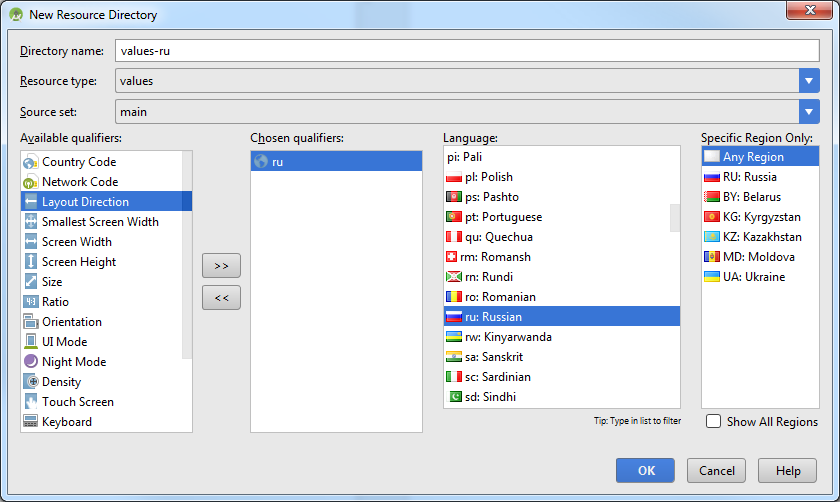


Рисунок 4 – Добавление локализации Россия

Созданная папка не будет отображаться. Чтобы добавить в неё файлы следует перейти в режим project (рис. 5).

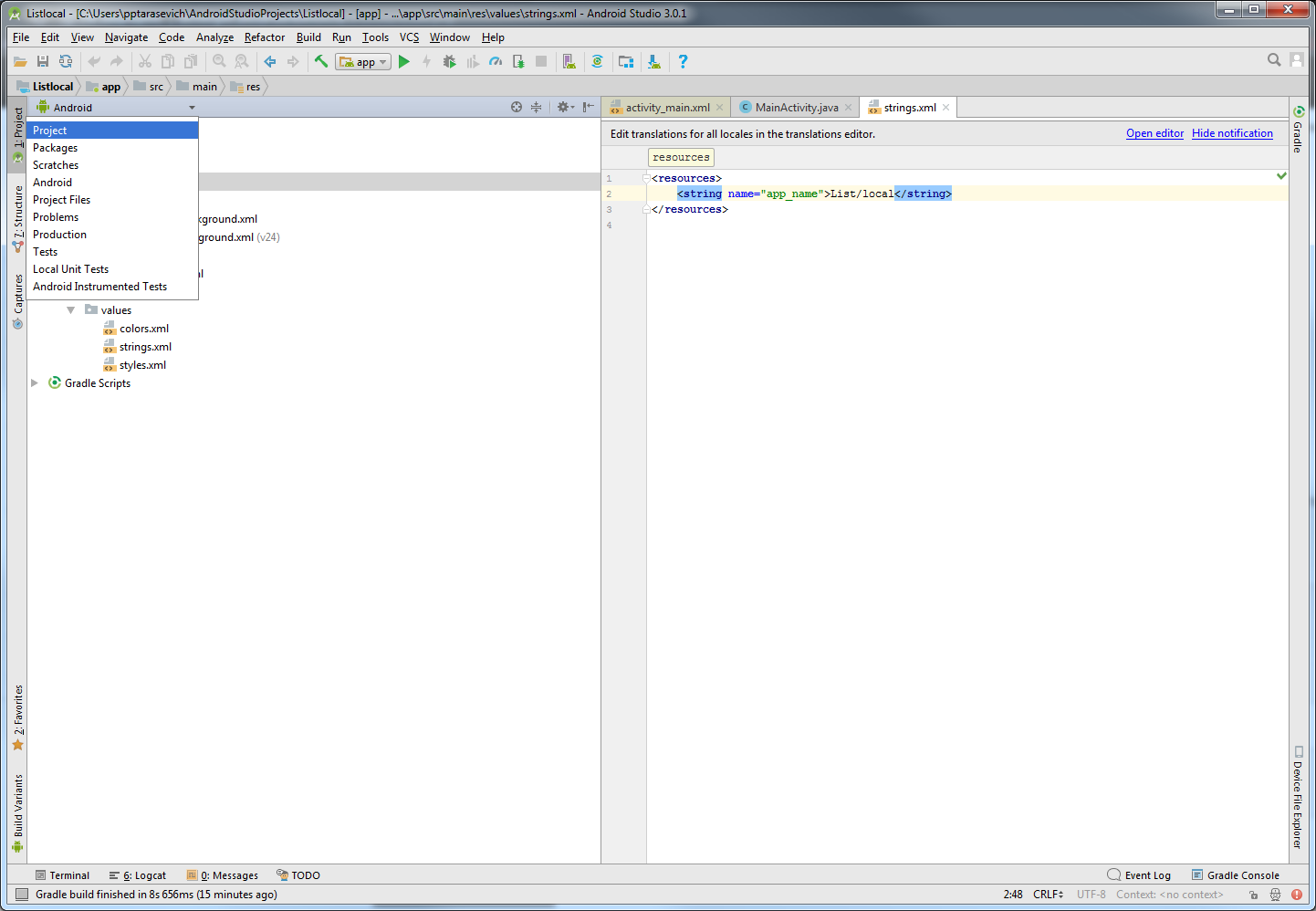


Рисунок 5 – Переключение вида

Теперь, раскроем папку с названием проекта и создадим в папке values-ru новый Values resource file.

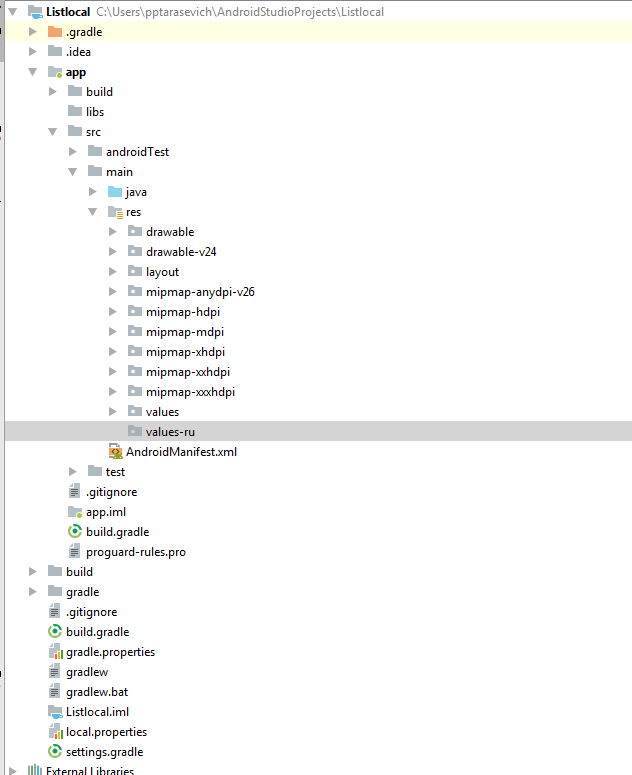


Рисунок 6 –Создание нового файла

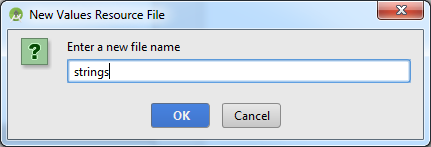


Рисунок 7 –Название файла ресурсов

Добавим строку.

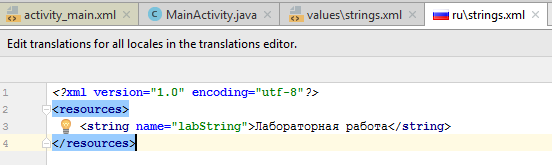


Рисунок 8 – Добавление ресурса

Теперь, если вернуться в режим Android, то вместо файла strings.xml появится папка, которая будет хранить уже два файла: один по умолчанию и второй, созданный только что.

Создадим теперь английский вариант того же самого файла и добавим строку с тем же именем.

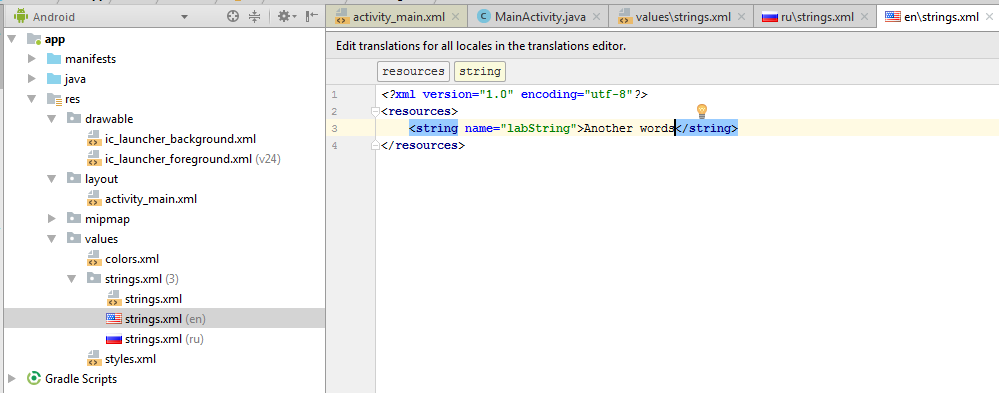


Рисунок 9 – Добавление английской локализации

Добавив в activity\_main.xml TextView проверим результаты.

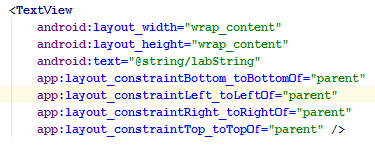


Рисунок 10 – Установка ресурса

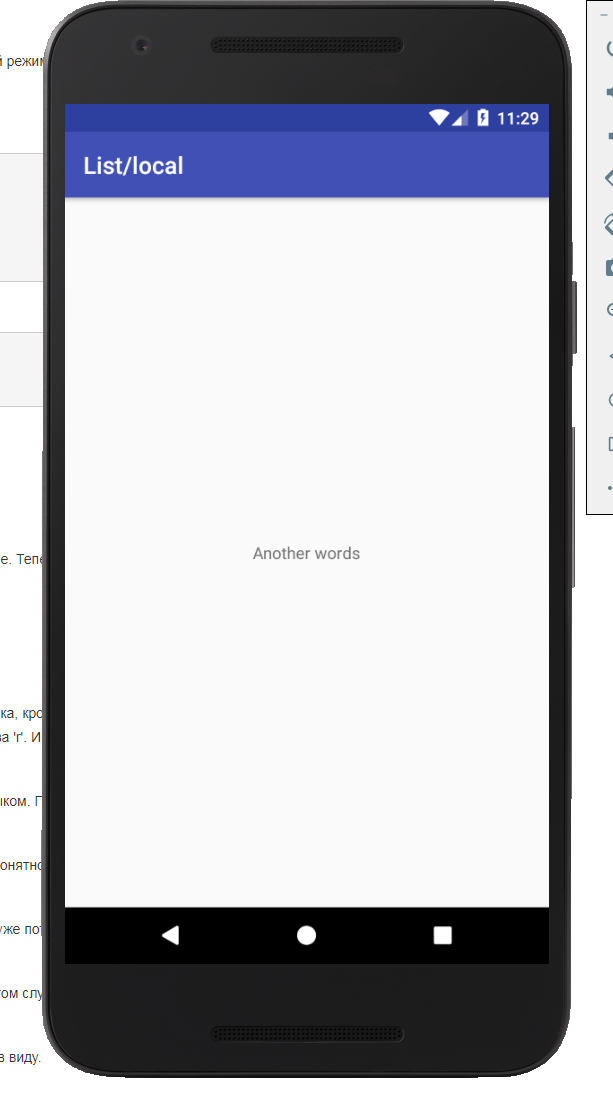


Рисунок 11 –Пример русского языка

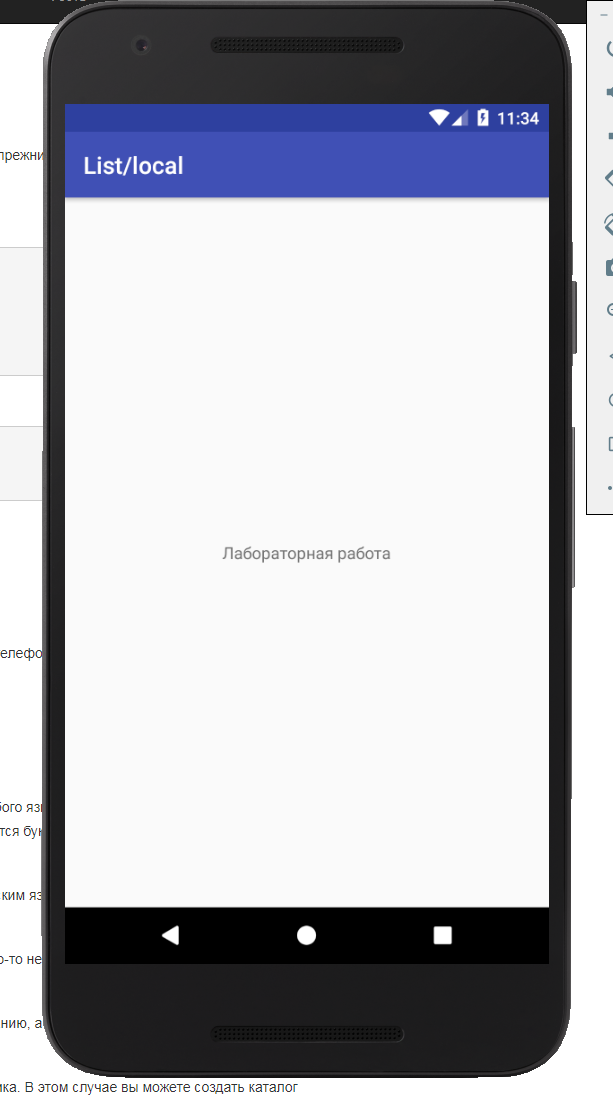


Рисунок 12 – Пример английского языка

Когда мы добавляем разметку в xml можно тут же и проверить, как будет отображаться строка для различных локализаций. Для этого в Preview есть пункт Language, где можно выбрать необходимую локализацию и просмотреть её отображение.

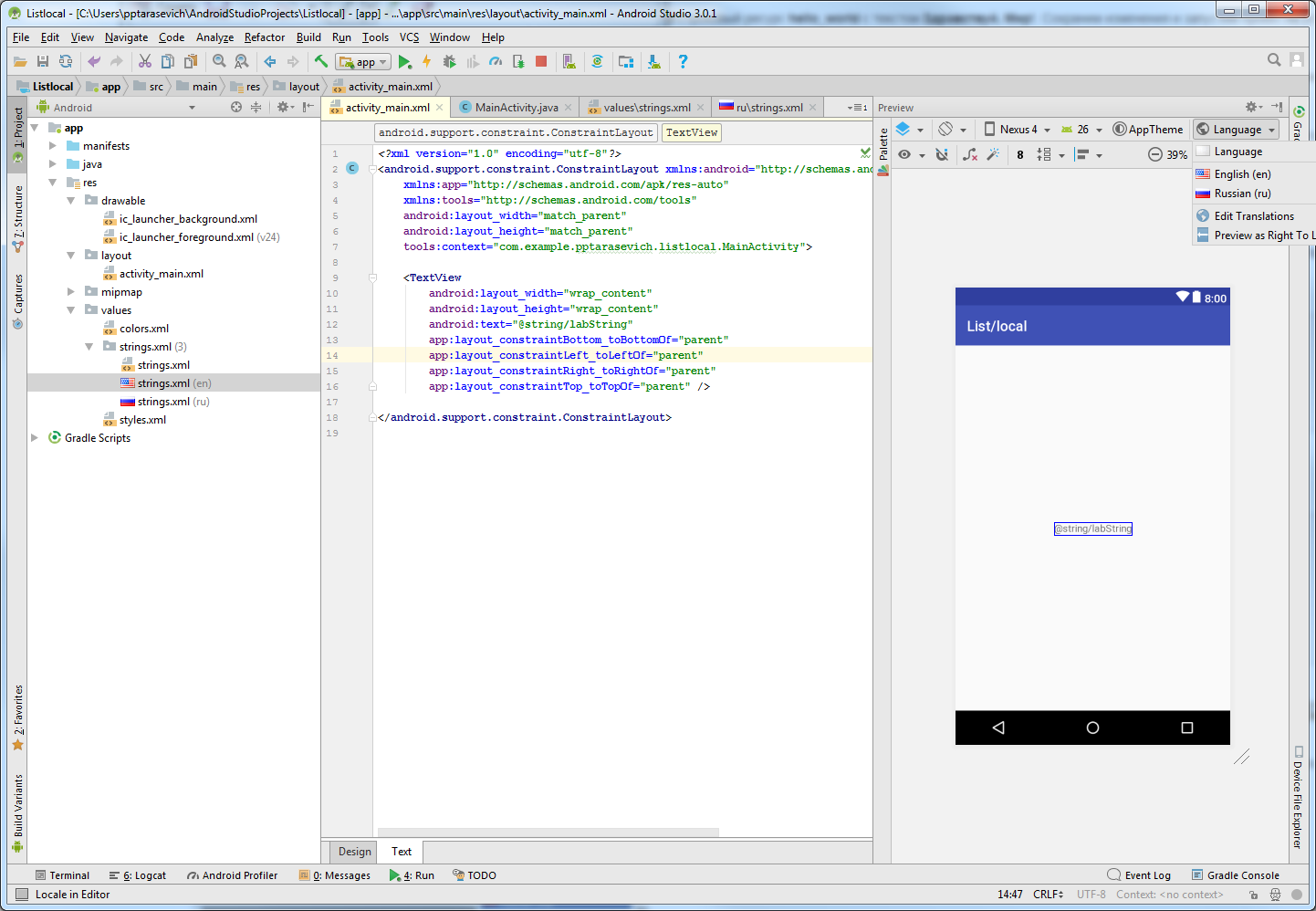


Рисунок 13 –Предпросмотр локализации

Теперь перейдём ко второму пункту лабораторной работы – списки. Списки используются повсеместно, будь то отображение контактов в телефоне, список товаров, список дел и тд. Отображать такой список с помощью TextView будет неудобно и уж тем более неправильно.

Для этой цели существует отличный виджет – ListView.

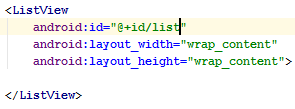


Рисунок 14 – Создание ListView

Смысл присутствия такого списка заключается в динамике. То есть, мы можем динамически удалять и добавлять элементы. В качестве посредника выступает ArrayAdapter, который содержит ссылку на источник объектов и ссылку на разметку. В качестве источника могут выступать обычные массивы, список, база данных. Чтобы заполнить ListView данными, например, из массива, используется следующий код.

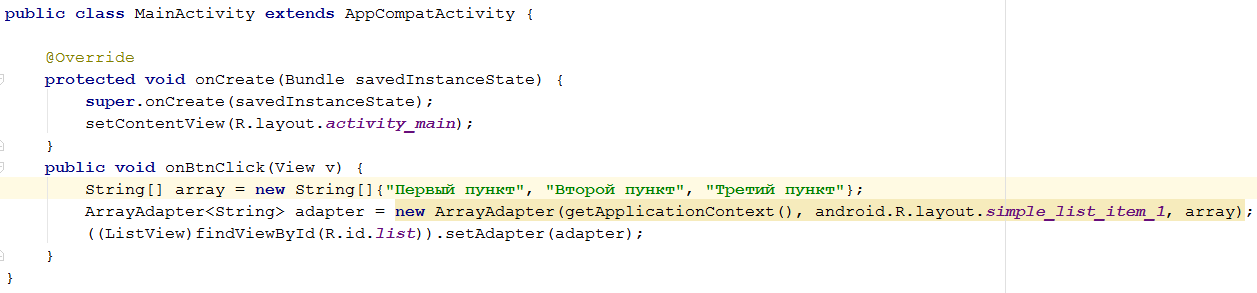


Рисунок 15 – Заполнение с использованием масива

Первый параметр – это контекст страницы, затем идёт ссылка на разметку (она есть по умолчанию), а затем ссылка на источник данных. В конце мы должны применить для ListView созданный адаптер.

Также, в качестве источника могут выступать ресурсы. Это очень полезно, если используется локализация в приложении. Пример такого использования представлен ниже.

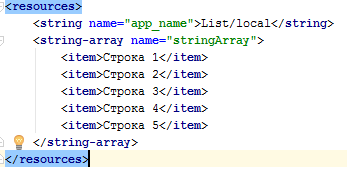


Рисунок 16 –Ввод ресурсов

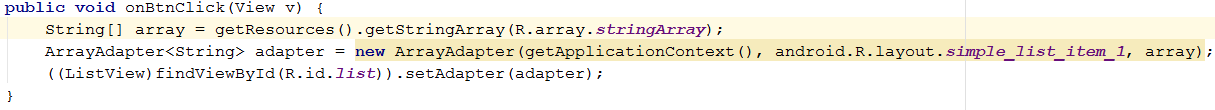


Рисунок 17 –Назначение ресурсов для ListView

Теперь нужно обработать выбор пункта из списка. Например, чтобы значение выделенной строки отображалось в TextView, изменим так, как показано на рисунке.

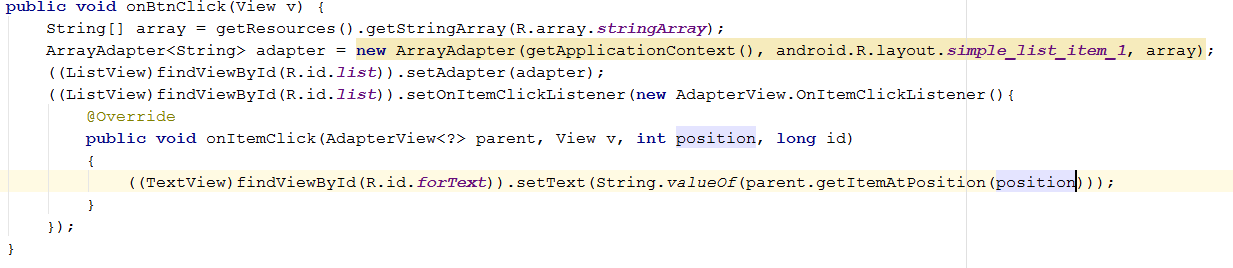


Рисунок 18 – Обработка нажатия

В качестве последнего пункта рассмотрим, какие есть формы ввода в Android Studio. Для взаимодействия с пользователем есть несколько способов. Самый простой способ – это кнопка. То есть, при нажатии на определённую кнопку будет срабатывать определённое действие. Тем не менее, иногда необходимо получить от пользователя неопределённую информацию. Для этого существуют поля ввода.

Самым распространённым пример служит EditText. EditText очень похож на input в html. Но, если в html абсолютно всё является input-ом, то здесь не так. Сюда, обычно, вносится некоторая текстовая информация. При этом, с помощью особого атрибута inputType можно ограничить символы, доступные пользователю. Существуют следующие ограничения:

* plainText (Значение по умолчанию) –любые символы можно вводить в данном поле;
* number – огранивает ввод только цифровыми значениями. То есть можно вставить только числа. Можно дополнительно ограничить только положительными числами (numberSigned) и возможность вводить числа с плавающей точкой (numberDecimal);
* textPassword – все символы будут заменены звёздочками (как в html). При этом можно ограничить для ввода только цифр, если добавить number (numberPassword);
* textEmailAddress – можно вносить данные только согласно шаблону (<name>@<subdomain>.<domain>);
* phone – используется для ввода телефона с возможностью добавления символов «\*» и «#»;
* time – можно вводить цифры и символ «:»;
* date – можно использовать для ввода даты (цифры, «.», «-»);

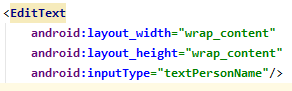


Рисунок 19 – Пример поля ввода

В html есть такой компонент, как textarea, который позволяет вводить текст, состоящий из нескольких строк. Здесь тоже есть такая возможность. Для этого в EditText inputType указывается как multiLine. Причём, выбрав данный тип, можно указать количество строк для отображение, указав атрибут lines.

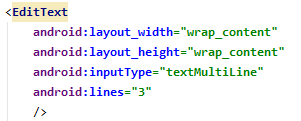


Рисунок 20 – Пример добавления многострочного поля ввода

Если Вы хотите помочь пользователю подсказкой, то можно указать так называемый placeholder. То есть некоторый текст, который будет отображаться, пока пользователь не начнёт вводить значение. В AndroidStudio данный атрибут носит название hint.

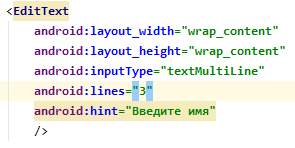




Рисунок 21 – Добавление замещающего текста

У текстовых полей есть атрибут imeOptions, с помощью которого настраиваются параметры для текущего метода ввода. Например, когда EditText получает фокус и отображается виртуальная клавиатура, эта клавиатура содержит кнопку «Next» (Далее), если атрибут imeOptions содержит значение actionNext. Если пользователь касается этой кнопки, фокус перемещается к следующему компоненту, который принимает пользовательский ввод. Если компонент EditText получает фокус и на виртуальной клавиатуре появляется кнопка «Done» (Готово), значит, использовался атрибут imeOptions со значением actionDone. Как только пользователь касается этой кнопки, система скрывает виртуальную клавиатуру.



Рисунок 22 – Дополнительные возможности использования клавиатуры

Атрибут maxLength позволяет ограничить количество символов, доступных для ввода в это поле.

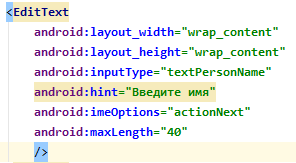


Рисунок 23 – Ограничение количества символов

Ранее можно было запретить пользователю вводить данные с помощью атрибута editable, который схож с enabled в html. Сейчас данный атрибут устарел. Если же у Вас есть необходимость запретить пользователю вводить некоторые данные в поле (например, пока не выберет, что согласен с условиями), можно использовать атрибут focusable.



Рисунок 24 – Запрет на редактирование

AndroidStudio не предоставляет атрибутов для валидации, как, например, атрибут pattern у input в html. Тем не менее, такая задача порой возникает у разработчиков. Для решения подобной задачи можно воспользоваться слушателем textChangeListener.

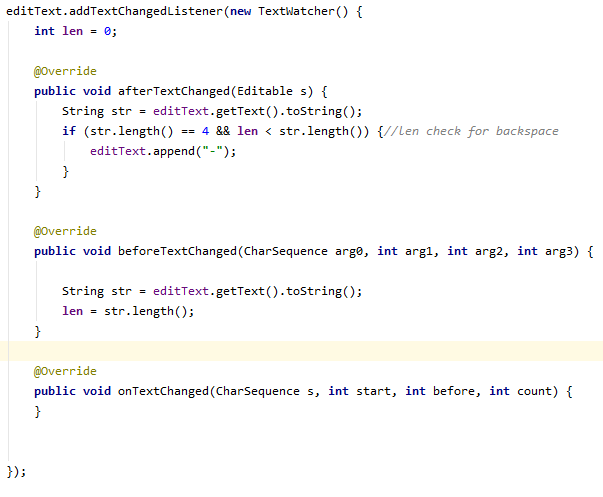


Рисунок 25 – Пример замены удалённых символов знаком «-»

В различные моменты времени мы можем получить значение того, что ввёл пользователь и обработать в соответствии с требованиями, которые мы указали для данного поля. Хорошо с этим справляются регулярные выражения.

Таблица 1.1 – регулярные выражения

|  |  |
| --- | --- |
| Выражение | Описание |
| \d [0-9] | Одна цифра от 0 до 9. |
| \D [^0-9] | Любой символ кроме цифры. |
| \s | Пробел. |
| [A-Z] | Только заглавная латинская буква. |
| [A-Za-z] | Только латинская буква в любом регистре. |
| [А-Яа-яЁё] | Только русская буква в любом регистре. |
| [A-Za-zА-Яа-яЁё] | Любая буква русского и латинского алфавита. |
| [0-9]{3} | Три цифры. |
| [A-Za-z]{6,} | Не менее шести латинских букв. |
| [0-9]{,3} | Не более трёх цифр. |
| [0-9]{5,10} | От пяти до десяти цифр. |
| ^[a-zA-Z]+$ | Любое слово на латинице. |
| ^[А-Яа-яЁё\s]+$ | Любое слово на русском включая пробелы. |
| ^[ 0-9]+$ | Любое число. |
| [0-9]{6} | Почтовый индекс. |
| \d+(,\d{2})? | Число в формате 1,34 (разделитель запятая). |
| \d+(\.\d{2})? | Число в формате 2.10 (разделитель точка). |
| \d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3} | IP-адрес |

Например

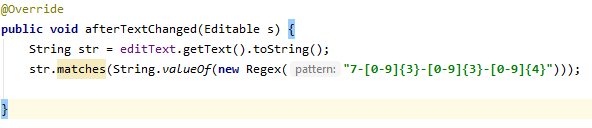


Рисунок 26 – добавление паттерна ввода номера телефона

Данный патерн осуществляет следующую валидацию: номер телефона должен начинаться с 7, затем символ дефис. После этого допустимо 3 символа цифры от 0 до 9. Затем, опять дефис и 3 любые цифры, дефис и ещё 4 цифры

Для ввода даты и времени помимо EditText можно использовать интерфейс, предлагаемые AndroidStudio. Данными решениями являются DatePicker и TimePicker. Они позволяют пользователю не вводить цифры вручную, а выбрать соответствующее значение с помощью визуального оформления в виде часов/календаря.

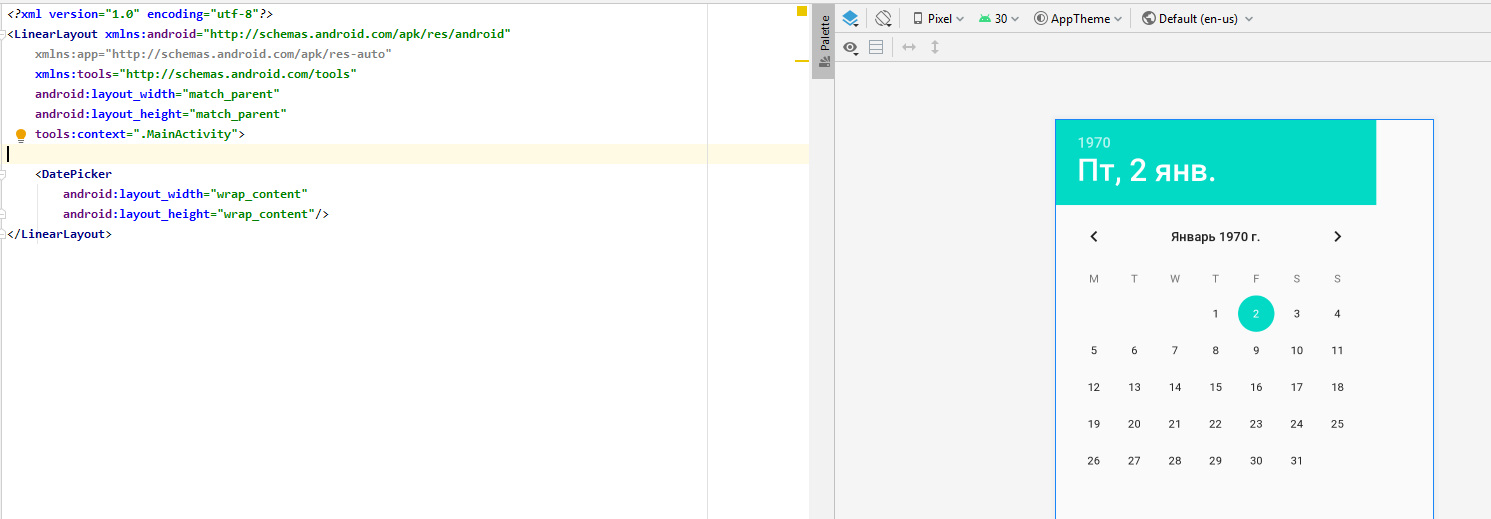


Рисунок 27 – Использование DatePicker

Вначале можно спрятать его, а по нажатию на определённую клавишу сделать видимым



Рисунок 28 – изменение атрибута видимости

Также, в AnroidStudio можно использовать такие компоненты, как Checkbox и RadioButton. C RadioButton Вы уже познакомились в лабораторной работе 5. Checkbox отличается только лишь тем, что можно одновременно выбрать несколько значений из списка.

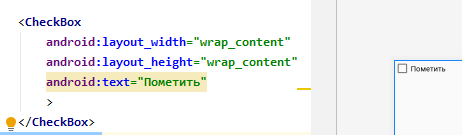


Рисунок 29 – Простейший пример Checkbox

Для выпадающего списка используется компонент Spinner:

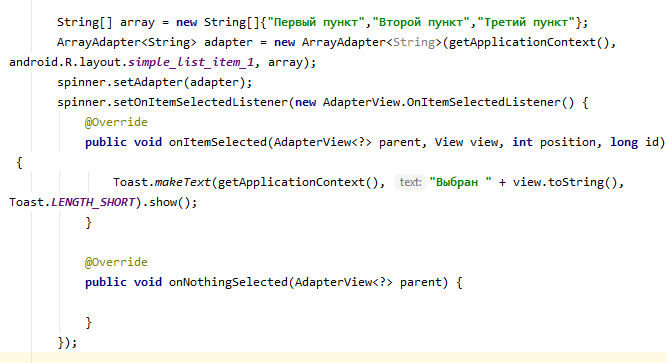


Рисунок 30 – Пример добавления выпадающего списка

**Задания**:

1. Создать список в файле ресурсов согласно варианту.
2. Создать локализованные ресурсы согласно варианту.
3. Добавить страницу для регистрации с использованием валидации:
   1. ФИО – только кириллические буквы, дефис и пробелы;
   2. Логин – только латиница;
   3. Email – валидный формат email-адрес;
   4. Номер телефона – в международной формате с вводом кода страны;
   5. Пароль;
   6. Повтор пароля – введенное значение должно совпадать с паролем;
   7. Дата рождения – валидная дата, не ранее 1900 года;
   8. Выпадающее меню для выбора из списка согласно варианту:
   9. Согласие на обработку персональных данных – должно быть отмечено.

В случае несоответствия любым требованиям выводится сообщение об ошибке и заносится в лог, поля с ошибками выделяются.

**Варианты:**

1. Список одногруппников (английский, русский).
2. Список имен котов (русский, казахский).
3. Список продуктов (русский, чешский).
4. Список преподавателей (русский, бурятский).
5. Список ингредиентов для пиццы (русский, итальянский).
6. Список ингредиентов для борща (русский, украинский).
7. Список заклинаний из Гарри Поттера (русский, английский).
8. Список периферийных устройств (русский, немецкий).
9. Список хоз. Инструментов (русский, вьетнамский).
10. Список мебели (русский, польский).

**Содержание отчета:** отчет по лабораторной работе должен быть выполнен в редакторе MSWord и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman, интервал – полуторный, поля левое – 3 см., правое – 1,5 см., верхнее и нижнее – 2 см. Абзацный отступ – 1,25. Текст должен быть выровнен по ширине.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись (Рисунок 1 – Создание подсистемы) и ссылку на него в тексте.

**Контрольные вопросы:**

**Список литературы, рекомендуемый к использованию по данной теме:**

1. Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. Основы интернет-технологий: учебное пособие / Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. – М.: Архангельск: ИПЦ САФУ, 2013 – 366 с.; То же [Электронный ресурс]. - URL:http://biblioclub.ru/index.php?page=book\_red&id=436379&sr=1
2. Соколова В. В. Разработка мобильных приложений: учебное пособие/ Соколова В. В. Томск: Издательство Томского политехнического университета, 2015 – 176с. То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book\_red&id=442808&sr=1