|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №9**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-12-22 | Солобай А.П. |
| **Проверил:**  Преподаватель | Степанов П.В. |

Москва 2024 г.

# Выполнение практической работы

Ссылка на GitHub со всеми работами: ***https://github.com/Eckorezze/Mobil***

1. Изучить применение адаптеров и списков. Реализовать пример связи

элемента ListView с источником данных с набором элементов TextView с

помощью одного из таких адаптеров - класса ArrayAdapter.

*Листинг 1 – кода примера*

|  |
| --- |
| String[] countries = { "Бразилия", "Аргентина", "Колумбия", "Чили", "Уругвай"}; ListView countriesList = findViewById(R.id.countriesList); ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  android.R.layout.simple\_list\_item\_1, countries); countriesList.setAdapter(adapter); |

Изображение выглядит как электроника, текст, снимок экрана, компьютер

Автоматически созданное описание

Рисунок 1 – тестирование кода

2. Реализовать пример с использованием ресурса string-array и ListView.

*Листинг 2 –код примера*

|  |
| --- |
| ListView countriesList = findViewById(R.id.countriesList); String[] countries = getResources().getStringArray(R.array.countries); ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  android.R.layout.simple\_list\_item\_1, countries); countriesList.setAdapter(adapter); |

Изображение выглядит как электроника, текст, снимок экрана, компьютер

Автоматически созданное описание

Рисунок 2 – тестирование кода

3. Реализовать выбор элемента в ListView.

*Листинг 3 – код примера*

|  |
| --- |
| ListView countriesList = findViewById(R.id.countriesList);  String[] countries = getResources().getStringArray(R.array.countries);  TextView selection = findViewById(R.id.selection);  ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  android.R.layout.simple\_list\_item\_1, countries);  countriesList.setAdapter(adapter);  countriesList.setOnItemClickListener((parent, v, position, id) -> {  String selectedItem = countries[position];  selection.setText(selectedItem);  }); } |

Изображение выглядит как электроника, текст, снимок экрана, мультимедиа

Автоматически созданное описание

Рисунок 3 – тестирование кода

4. Реализовать множественный выбор в списке.

*Листинг 4 – код примера*

|  |
| --- |
| TextView selection = findViewById(R.id.selection); ListView countriesList = findViewById(R.id.countriesList); ArrayAdapter<String> adapter = new ArrayAdapter(this,  android.R.layout.simple\_list\_item\_multiple\_choice, countries); countriesList.setAdapter(adapter); countriesList.setOnItemClickListener((parent, v, position, id) -> {  SparseBooleanArray selected = countriesList.getCheckedItemPositions();  StringBuilder selectedItems = new StringBuilder();  for (int i = 0; i < countries.length; i++) {  if (selected.get(i))  selectedItems.append(countries[i]).append(",");  }  *// установка текста элемента TextView* selection.setText("Выбрано: " + selectedItems); }); |

Изображение выглядит как электроника, текст, снимок экрана, компьютер

Автоматически созданное описание

Рисунок 4 – тестирование кода

5. Реализовать добавление и удаление в ArrayAdapter и ListView.

*Листинг 5 – код примера*

|  |
| --- |
| Collections.addAll(users, "Tom", "Bob", "Sam", "Alice");  usersList = findViewById(R.id.usersList);  adapter = new ArrayAdapter<>(this,  android.R.layout.simple\_list\_item\_multiple\_choice, users);  usersList.setAdapter(adapter);  usersList.setOnItemClickListener((parent, v, position, id) -> {  String user = adapter.getItem(position);  if(usersList.isItemChecked(position))  selectedUsers.add(user);  else  selectedUsers.remove(user);  }); } public void add(View view){  EditText userName = findViewById(R.id.userName);  String user = userName.getText().toString();  if(!user.isEmpty()){  adapter.add(user);  userName.setText("");  adapter.notifyDataSetChanged();  } } public void remove(View view){  for(int i=0; i< selectedUsers.size();i++){  adapter.remove(selectedUsers.get(i));  }  usersList.clearChoices();  selectedUsers.clear();  adapter.notifyDataSetChanged(); } |

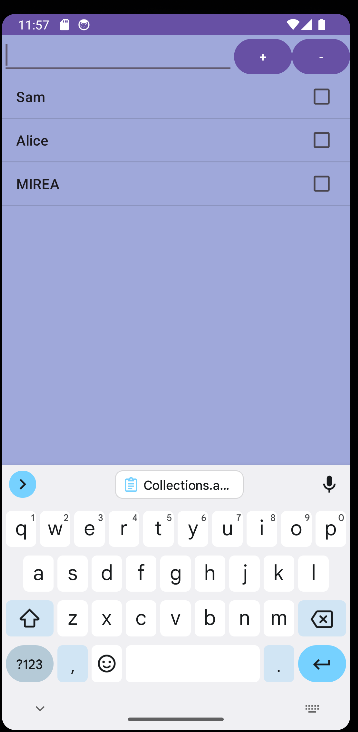


Рисунок 5 – тестирование кода

6. Реализовать расширение списков и создание адаптера.

*Листинг 6 – код в файле PracticeActivity*

|  |
| --- |
| setInitialData();  countriesList = findViewById(R.id.countriesList);  StateAdapter stateAdapter = new StateAdapter(this, R.layout.list\_item,  states);  countriesList.setAdapter(stateAdapter);  AdapterView.OnItemClickListener itemListener = (parent, v, position, id) -> {  State selectedState = (State) parent.getItemAtPosition(position);  Toast.makeText(getApplicationContext(), "Был выбран пункт " +  selectedState.getName(),  Toast.LENGTH\_SHORT).show();  };  countriesList.setOnItemClickListener(itemListener); }  private void setInitialData() {  states.add(new State("Бразилия", "Бразилиа", R.drawable.brazilia));  states.add(new State("Аргентина", "Буэнос-Айрес", R.drawable.argentina));  states.add(new State("Колумбия", "Богота", R.drawable.columbia));  states.add(new State("Уругвай", "Монтевидео", R.drawable.uruguai));  states.add(new State("Чили", "Сантьяго", R.drawable.chile)); } |

*Листинг 7 – код в файле State*

|  |
| --- |
| public class State {  private String name;  private String capital;  private int flagResource;  public State(String name, String capital, int flag){  this.name=name;  this.capital=capital;  this.flagResource=flag;  }  public String getName() {  return this.name;  }  public void setName(String name) {  this.name = name;  }  public String getCapital() {  return this.capital;  }  public void setCapital(String capital) {  this.capital = capital;  }  public int getFlagResource() {  return this.flagResource;  }  public void setFlagResource(int flagResource) {  this.flagResource = flagResource;  } } |

*Листинг 8 – код в файле StateAdapter*

|  |
| --- |
| public class StateAdapter extends ArrayAdapter<State> {  private LayoutInflater inflater;  private int layout;  private List<State> states;  public StateAdapter(Context context, int resource, List<State> states) {  super(context, resource, states);  this.states = states;  this.layout = resource;  this.inflater = LayoutInflater.from(context);  }  @NonNull  public View getView(int position, View convertView, ViewGroup parent) {  @SuppressLint("ViewHolder") View view=inflater.inflate(this.layout, parent, false);  ImageView flagView = view.findViewById(R.id.flag);  TextView nameView = view.findViewById(R.id.name);  TextView capitalView = view.findViewById(R.id.capital);  State state = states.get(position);  flagView.setImageResource(state.getFlagResource());  nameView.setText(state.getName());  capitalView.setText(state.getCapital());  return view;  } } |

Изображение выглядит как электроника, текст, снимок экрана, мультимедиа

Автоматически созданное описание

Рисунок 6 – тестирование кода

7. Реализовать оптимизацию адаптера и применение View Holder.

*Листинг 9 – код примера*

|  |
| --- |
| public class StateAdapter extends ArrayAdapter<State> {  private final LayoutInflater inflater;  private final int layout;  private final List<State> states;  public StateAdapter(Context context, int resource, List<State> states) {  super(context, resource, states);  this.states = states;  this.layout = resource;  this.inflater = LayoutInflater.from(context);  }  @NonNull  public View getView(int position, View convertView, @NonNull ViewGroup parent) {  ViewHolder viewHolder;  if(convertView==null){  convertView = inflater.inflate(this.layout, parent, false);  viewHolder = new ViewHolder(convertView);  convertView.setTag(viewHolder);  }  else{  viewHolder = (ViewHolder) convertView.getTag();  }  State state = states.get(position);  viewHolder.imageView.setImageResource(state.getFlagResource());  viewHolder.nameView.setText(state.getName());  viewHolder.capitalView.setText(state.getCapital());  return convertView;  }  private static class ViewHolder {  final ImageView imageView;  final TextView nameView, capitalView;  ViewHolder(View view){  imageView = view.findViewById(R.id.flag);  nameView = view.findViewById(R.id.name);  capitalView = view.findViewById(R.id.capital);  }  } } |

Изображение выглядит как электроника, текст, снимок экрана, Операционная система

Автоматически созданное описание

Рисунок 7 – тестирование кода

8. Реализовать сложный список с кнопками.

*Листинг 10 – код в файле PracticeActivity*

|  |
| --- |
| ArrayList<Product> products = new ArrayList<Product>(); products.size(); products.add(new Product("Кириешки", "шт.")); products.add(new Product("Пиво", "шт.")); products.add(new Product("Сигареты", "шт.")); products.add(new Product("Энергетик", "шт.")); ListView productList = findViewById(R.id.productList); ProductAdapter adapter = new ProductAdapter(this, R.layout.list\_item,  products); productList.setAdapter(adapter); |

*Листинг 11 – код в файле ProductAdapter*

|  |
| --- |
| class ProductAdapter extends ArrayAdapter<Product> {  private final LayoutInflater inflater;  private final int layout;  private final ArrayList<Product> productList;   ProductAdapter(Context context, int resource, ArrayList<Product> products) {  super(context, resource, products);  this.productList = products;  this.layout = resource;  this.inflater = LayoutInflater.from(context);  }   @SuppressLint("SetTextI18n")  @NonNull  public View getView(int position, View convertView, @NonNull ViewGroup parent) {  final ViewHolder viewHolder;  if (convertView == null) {  convertView = inflater.inflate(this.layout, parent, false);  viewHolder = new ViewHolder(convertView);  convertView.setTag(viewHolder);  } else {  viewHolder = (ViewHolder) convertView.getTag();  }  final Product product = productList.get(position);  viewHolder.nameView.setText(product.getName());  viewHolder.countView.setText(product.getCount() + " " + product.getUnit());  viewHolder.removeButton.setOnClickListener(v -> {  int count = product.getCount() - 1;  if (count < 0) count = 0;  product.setCount(count);  viewHolder.countView.setText(count + " " + product.getUnit());  });  viewHolder.addButton.setOnClickListener(v -> {  int count = product.getCount() + 1;  product.setCount(count);  viewHolder.countView.setText(count + " " + product.getUnit());  });  return convertView;  }   private static class ViewHolder {  final Button addButton, removeButton;  final TextView nameView, countView;   ViewHolder(View view) {  addButton = view.findViewById(R.id.addButton);  removeButton = view.findViewById(R.id.removeButton);  nameView = view.findViewById(R.id.nameView);  countView = view.findViewById(R.id.countView);  }  }  } |

*Листинг 12 – код в файле Product*

|  |
| --- |
| public class Product {  private String name;  private int count;  private final String unit;  Product(String name, String unit){  this.name = name;  this.count=0;  this.unit = unit;  }  public String getUnit() {  return this.unit;  }  public void setCount(int count) {  this.count = count;  }  public int getCount() {  return count;  }  public void setName(String name){  this.name = name;  }  public String getName(){  return this.name;  } } |

Изображение выглядит как снимок экрана, текст, мультимедиа, Операционная система

Автоматически созданное описание

Рисунок 8 – тестирование кода

9. Реализовать выпадающий список Spinner.

*Листинг 13 – код в файле PracticeActivity*

|  |
| --- |
| String[] countries = {"Бразилия", "Аргентина", "Колумбия", "Чили", "Уругвай"}; Spinner spinner = findViewById(R.id.spinner); ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  android.R.layout.simple\_spinner\_item, countries);  adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item); spinner.setAdapter(adapter); |

*Листинг 14 – код в файле practice\_activity*

|  |
| --- |
| <Spinner  android:id="@+id/spinner"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintTop\_toTopOf="parent" /> |

Изображение выглядит как текст, снимок экрана, компьютер, Операционная система

Автоматически созданное описание

Рисунок 9 – тестирование кода

10.Реализовать обработку выбора элемента используя слушатель

OnItemSelectedListener.

*Листинг 15 – код примера*

|  |
| --- |
| String[] countries = { "Бразилия", "Аргентина", "Колумбия", "Чили", "Уругвай"}; TextView selection = findViewById(R.id.selection); Spinner spinner = findViewById(R.id.spinner); ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple\_spinner\_item, countries);  adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item); spinner.setAdapter(adapter); AdapterView.OnItemSelectedListener itemSelectedListener = new  AdapterView.OnItemSelectedListener() {  @Override  public void onItemSelected(AdapterView<?> parent, View view, int  position, long id) {  String item = (String)parent.getItemAtPosition(position);  selection.setText(item);  }  @Override  public void onNothingSelected(AdapterView<?> parent) {  }  }; spinner.setOnItemSelectedListener(itemSelectedListener); |

Изображение выглядит как снимок экрана, компьютер, текст, мультимедиа

Автоматически созданное описание

Рисунок 10 – тестирование кода

11.Реализовать виджет автодополнения AutoCompleteTextView.

*Листинг 16 – код в файле PracticeActivity*

|  |
| --- |
| String[] cities = {"Москва", "Самара", "Вологда", "Волгоград", "Саратов", "Воронеж"}; AutoCompleteTextView autoCompleteTextView =  findViewById(R.id.autocomplete); ArrayAdapter<String> adapter = new ArrayAdapter(this, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item, cities); autoCompleteTextView.setAdapter(adapter); |

*Листинг 17 – код в файле practice\_activity*

|  |
| --- |
| <AutoCompleteTextView  android:id="@+id/autocomplete"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:completionHint="Введите город"  android:completionThreshold="1"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  /> |

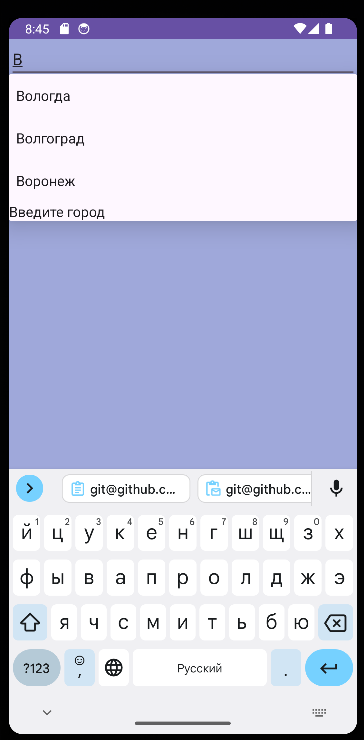


Рисунок 11 – тестирование кода

12.Реализовать виджет MultiAutoCompleteTextView.

*Листинг 18 – код примера*

|  |
| --- |
| String[] cities = {"Москва", "Самара", "Вологда", "Волгоград", "Саратов", "Воронеж"}; MultiAutoCompleteTextView autoCompleteTextView = findViewById(R.id.autocomplete); ArrayAdapter<String> adapter = new ArrayAdapter(this,  androidx.constraintlayout.widget.R.layout.support\_simple\_spinner\_dropdown\_item, cities); autoCompleteTextView.setAdapter(adapter); autoCompleteTextView.setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer()); |

Изображение выглядит как электроника, снимок экрана, текст, Электронное устройство

Автоматически созданное описание

Рисунок 12 – тестирование кода

13.Реализовать пример с элементом GridView.

*Листинг 19 – тестирование кода примера*

|  |
| --- |
| String[] countries = { "Бразилия", "Аргентина", "Чили", "Колумбия", "Уругвай"}; GridView countriesList = findViewById(R.id.gridview); ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  android.R.layout.simple\_list\_item\_1, countries); countriesList.setAdapter(adapter); AdapterView.OnItemClickListener itemListener = (parent, view1, position, id) -> Toast.makeText(getApplicationContext(),"Вы выбрали "  + parent.getItemAtPosition(position).toString(),  Toast.LENGTH\_SHORT).show(); countriesList.setOnItemClickListener(itemListener); |

Изображение выглядит как электроника, текст, снимок экрана, мультимедиа

Автоматически созданное описание

Рисунок 13 – тестирование кода

14.Реализовать пример с элементом RecyclerView.

*Листинг 20 – код в файле PracticeActivity*

|  |
| --- |
| ArrayList<State> states = new ArrayList<State>();  *// начальная инициализация списка* setInitialData();  RecyclerView recyclerView = findViewById(R.id.list);  *// создаем адаптер* StateAdapter adapter = new StateAdapter(this, states);  *// устанавливаем для списка адаптер* recyclerView.setAdapter(adapter); } private void setInitialData(){  states.add(new State ("Бразилия", "Бразилиа", R.drawable.brazilia));  states.add(new State ("Аргентина", "Буэнос-Айрес", R.drawable.argentina));  states.add(new State ("Колумбия", "Богота", R.drawable.columbia));  states.add(new State ("Уругвай", "Монтевидео", R.drawable.uruguai));  states.add(new State ("Чили", "Сантьяго", R.drawable.chile)); } |

*Листинг 21 – код в файле StateAdapter*

|  |
| --- |
| public class StateAdapter extends  RecyclerView.Adapter<StateAdapter.ViewHolder>{  private final LayoutInflater inflater;  private final List<State> states;  StateAdapter(Context context, List<State> states) {  this.states = states;  this.inflater = LayoutInflater.from(context);  }  @NonNull  @Override  public StateAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int  viewType) {  View view = inflater.inflate(R.layout.list\_item, parent, false);  return new ViewHolder(view);  }  @Override  public void onBindViewHolder(StateAdapter.ViewHolder holder, int position) {  State state = states.get(position);  holder.flagView.setImageResource(state.getFlagResource());  holder.nameView.setText(state.getName());  holder.capitalView.setText(state.getCapital());  }  @Override  public int getItemCount() {  return states.size();  }  public static class ViewHolder extends RecyclerView.ViewHolder {  final ImageView flagView;  final TextView nameView, capitalView;  ViewHolder(View view){  super(view);  flagView = view.findViewById(R.id.flag);  nameView = view.findViewById(R.id.name);  capitalView = view.findViewById(R.id.capital);  }  } } |

Изображение выглядит как электроника, снимок экрана, Операционная система, мультимедиа

Автоматически созданное описание

Рисунок 14 – тестирование кода

15.Реализовать обработку выбора элемента в RecyclerView.

*Листинг 22 – код в файле PracticeActivity*

|  |
| --- |
| setInitialData();  RecyclerView recyclerView = findViewById(R.id.list);  StateAdapter.OnStateClickListener stateClickListener = (state, position) -> Toast.makeText(getApplicationContext(), "Был выбран пункт " +  state.getName(),  Toast.LENGTH\_SHORT).show();  StateAdapter adapter = new StateAdapter(this, states, stateClickListener);  recyclerView.setAdapter(adapter); } private void setInitialData(){  states.add(new State ("Бразилия", "Бразилиа", R.drawable.brazilia));  states.add(new State ("Аргентина", "Буэнос-Айрес", R.drawable.argentina));  states.add(new State ("Колумбия", "Богота", R.drawable.columbia));  states.add(new State ("Уругвай", "Монтевидео", R.drawable.uruguai));  states.add(new State ("Чили", "Сантьяго", R.drawable.chile)); } |

*Листинг 23 – код в файле StateAdapter*

|  |
| --- |
| public class StateAdapter extends  RecyclerView.Adapter<StateAdapter.ViewHolder>{  interface OnStateClickListener{  void onStateClick(State state, int position);  }  private final OnStateClickListener onClickListener;  private final LayoutInflater inflater;  private final List<State> states;  StateAdapter(Context context, List<State> states, OnStateClickListener  onClickListener) {  this.onClickListener = onClickListener;  this.states = states;  this.inflater = LayoutInflater.from(context);  }  @NonNull  @Override  public StateAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int  viewType) {  View view = inflater.inflate(R.layout.list\_item, parent, false);  return new ViewHolder(view);  }  @Override  public void onBindViewHolder(StateAdapter.ViewHolder holder, int position) {  State state = states.get(position);  holder.flagView.setImageResource(state.getFlagResource());  holder.nameView.setText(state.getName());  holder.capitalView.setText(state.getCapital());  holder.itemView.setOnClickListener(v -> onClickListener.onStateClick(state, position));  }  @Override  public int getItemCount() {  return states.size();  }  public static class ViewHolder extends RecyclerView.ViewHolder {  final ImageView flagView;  final TextView nameView, capitalView;  ViewHolder(View view){  super(view);  flagView = view.findViewById(R.id.flag);  nameView = view.findViewById(R.id.name);  capitalView = view.findViewById(R.id.capital);  }  } } |

Изображение выглядит как электроника, снимок экрана, текст, мультимедиа

Автоматически созданное описание

Рисунок 15 – тестирование кода

# Вывод

В ходе работы мне удалось научиться работать с различными адаптерами и списками, а также с их оптимизацией.