****

**移动终端程序设计**

**实验报告**

|  |  |
| --- | --- |
| 班 级： | **专业210X** |
| 学 号： | **XXXXXXXX** |
| 姓 名： | **XX/XXX/XXXX** |

**东北大学秦皇岛分校计算机与通信工程学院**

**2024**

|  |  |
| --- | --- |
| **【实验编号】** | **1** |
| **【实验名称】** | **基本页面的设计与实现 学时：2** |
| **【实验目的】**  1）熟悉安卓开发环境  2）掌握基本控件的使用  3）掌握控件的事件响应机制  4）运行一个简单的UI界面  **【实验内容】**  使用Android Studio，编写一个简单的UI界面。  **【实验代码】**  MainActivity.java：   1. public class MainActivity extends AppCompatActivity { 2. private List<Msg> msgList = new ArrayList<Msg>(); 3. private EditText inputText; 4. private Button send; 5. private RecyclerView msgRecyclerView; 6. private MsgAdapter adapter; 7. @Override 8. protected void onCreate(Bundle savedInstanceState) { 9. super.onCreate(savedInstanceState); 10. EdgeToEdge.enable(this); 11. setContentView(R.layout.activity\_main); 12. initMsgs(); *// 初始化消息数据* 13. inputText = (EditText) findViewById(R.id.input\_text); 14. send = (Button) findViewById(R.id.send); 15. msgRecyclerView = (RecyclerView) findViewById(R.id.msg\_recycler\_view); 16. LinearLayoutManager layoutManager = new LinearLayoutManager(this); 17. msgRecyclerView.setLayoutManager(layoutManager); 18. adapter = new MsgAdapter(msgList); 19. msgRecyclerView.setAdapter(adapter); 20. send.setOnClickListener(new View.OnClickListener() { 21. @Override 22. public void onClick(View v) { 23. String content = inputText.getText().toString(); 24. if (!"".equals(content)) { 25. Msg msg = new Msg(content, Msg.TYPE\_SENT); 26. msgList.add(msg); 27. adapter.notifyItemInserted(msgList.size() - 1); 28. msgRecyclerView.scrollToPosition(msgList.size() - 1); *// 将ListView定位到最后一行* 29. inputText.setText(""); *// 清空输入框中的内容* 30. } 31. } 32. }); 33. } 34. @Override 35. protected void onResume() { 36. super.onResume(); 37. adjustInputLayoutHeight(); 38. } 39. private void initMsgs() { 40. Msg msg1 = new Msg("你好", Msg.TYPE\_RECEIVED); 41. msgList.add(msg1); 42. Msg msg2 = new Msg("你是哪位?", Msg.TYPE\_SENT); 43. msgList.add(msg2); 44. Msg msg3 = new Msg("我是你同学. ", Msg.TYPE\_RECEIVED); 45. msgList.add(msg3); 46. } 47. private void adjustInputLayoutHeight() { 48. LinearLayout inputLayout = findViewById(R.id.input\_layout); 49. Resources resources = getResources(); 50. int resourceId = resources.getIdentifier("navigation\_bar\_height", "dimen", "android"); 51. if (resourceId > 0) { 52. int navigationHeight = resources.getDimensionPixelSize(resourceId); 53. inputLayout.setPadding(0, 0, 0, navigationHeight); 54. } 55. } 56. }   MsgAdapter.java代码：   1. public class Msg { 2. public static final int TYPE\_RECEIVED = 0; 3. public static final int TYPE\_SENT = 1; 4. private String content; 5. private int type; 6. public Msg(String content, int type) { 7. this.content = content; 8. this.type = type; 9. } 10. public String getContent() { 11. return content; 12. } 13. public int getType() { 14. return type; 15. } 16. }     Msg.java代码：   1. public class MsgAdapter extends RecyclerView.Adapter<MsgAdapter.ViewHolder> { 2. private List<Msg> mMsgList; 3. static class ViewHolder extends RecyclerView.ViewHolder { 4. LinearLayout leftLayout; 5. LinearLayout rightLayout; 6. TextView leftMsg; 7. TextView rightMsg; 8. public ViewHolder(View view) { 9. super(view); 10. leftLayout = (LinearLayout) view.findViewById(R.id.left\_layout); 11. rightLayout = (LinearLayout) view.findViewById(R.id.right\_layout); 12. leftMsg = (TextView) view.findViewById(R.id.left\_msg); 13. rightMsg = (TextView) view.findViewById(R.id.right\_msg); 14. } 15. } 16. public MsgAdapter(List<Msg> msgList) { 17. mMsgList = msgList; 18. } 19. public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) { 20. View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.msg\_item, parent, false); 21. return new ViewHolder(view); 22. } 23. public void onBindViewHolder(ViewHolder holder, int position) { 24. Msg msg = mMsgList.get(position); 25. if (msg.getType() == Msg.TYPE\_RECEIVED) { 26. holder.leftLayout.setVisibility(View.VISIBLE); 27. holder.rightLayout.setVisibility(View.GONE); 28. holder.leftMsg.setText(msg.getContent()); 29. } else if(msg.getType() == Msg.TYPE\_SENT) { 30. holder.rightLayout.setVisibility(View.VISIBLE); 31. holder.leftLayout.setVisibility(View.GONE); 32. holder.rightMsg.setText(msg.getContent()); 33. } 34. } 35. public int getItemCount() { 36. return mMsgList.size(); 37. } 38. }     activity\_main.xml代码：   1. <?xml version="1.0" encoding="utf-8"?> 2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" 3. android:orientation="vertical" 4. android:layout\_width="match\_parent" 5. android:layout\_height="match\_parent" 6. android:fitsSystemWindows="true" 7. android:background="#d8c0e8"> 8. <androidx.recyclerview.widget.RecyclerView 9. android:id="@+id/msg\_recycler\_view" 10. android:layout\_width="match\_parent" 11. android:layout\_weight="1" 12. android:layout\_height="0dp" /> 13. <LinearLayout 14. android:id="@+id/input\_layout" 15. android:layout\_width="match\_parent" 16. android:layout\_height="wrap\_content" > 17. <EditText 18. android:id="@+id/input\_text" 19. android:layout\_width="0dp" 20. android:layout\_height="wrap\_content" 21. android:layout\_weight="1" 22. android:hint="请留言" 23. android:maxLines="2" /> 24. <Button 25. android:id="@+id/send" 26. android:layout\_width="wrap\_content" 27. android:layout\_height="wrap\_content" 28. android:text="发送" /> 29. </LinearLayout> 30. </LinearLayout>   msg\_item.xml代码：   1. <?xml version="1.0" encoding="utf-8"?> 2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" 3. android:orientation="vertical" 4. android:layout\_width="match\_parent" 5. android:layout\_height="wrap\_content" 6. android:padding="10dp"> 7. <LinearLayout 8. android:id="@+id/left\_layout" 9. android:layout\_width="wrap\_content" 10. android:layout\_height="wrap\_content" 11. android:layout\_gravity="left" 12. android:background="@drawable/message\_left"> 13. <TextView 14. android:id="@+id/left\_msg" 15. android:layout\_width="wrap\_content" 16. android:layout\_height="wrap\_content" 17. android:layout\_gravity="center" 18. android:layout\_margin="10dp" 19. android:textColor="#fff" /> 20. </LinearLayout> 21. <LinearLayout 22. android:id="@+id/right\_layout" 23. android:layout\_width="wrap\_content" 24. android:layout\_height="wrap\_content" 25. android:layout\_gravity="right" 26. android:background="@drawable/message\_right"> 27. <TextView 28. android:id="@+id/right\_msg" 29. android:layout\_width="wrap\_content" 30. android:layout\_height="wrap\_content" 31. android:layout\_gravity="center" 32. android:layout\_margin="10dp" /> 33. </LinearLayout> 34. </LinearLayout>   **【实验结果】**    **【实验心得体会】**  通过本次实验，学习了如何配置安卓开发环境，学习了各种控件的布局与使用，掌握了控件的事件响应机制，搭建了类似微信的聊天界面UI，通过调试UI界面的具体配置，对编写UI界面有了更深的理解。 | |

|  |  |
| --- | --- |
| **【实验编号】** | **2** |
| **【实验名称】** | **短信的发送与状态跟踪 学时：2** |
| **【实验目的】**  1) 了解服务的相关概念及运行机制  2) 实现短信的发送与数据的记录  **【实验内容】**  设计一个程序，可以发送、接受短信，将发送和接受的短信插入短信数据库  运行程序，查看运行效果。  **【实验代码】**  MainActivity.java:   1. public class MainActivity extends AppCompatActivity { 2. private Button button; 3. @Override 4. protected void onCreate(Bundle savedInstanceState) { 5. super.onCreate(savedInstanceState); 6. EdgeToEdge.enable(this); 7. setContentView(R.layout.activity\_main); 8. ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> { 9. Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars()); 10. v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom); 11. return insets; 12. }); 13. button = (Button) this.findViewById(R.id.send); 14. button.setOnClickListener(new View.OnClickListener() { 15. @RequiresApi(api = Build.VERSION\_CODES.O) 16. @Override 17. public void onClick(View view) { 18. send(view); 19. } 20. }); 21. } 22. @RequiresApi(api = Build.VERSION\_CODES.O) 23. public void send(View view) { 24. EditText num = (EditText) this.findViewById(R.id.phone\_number); 25. EditText msg = (EditText) this.findViewById(R.id.input\_text); 26. String str\_num = num.getText().toString();*// 得到电话号码* 27. String str\_msg = msg.getText().toString();*// 得到短信内容* 28. String texts = "短信发送器，短信内容：" + str\_msg; 29. String SENT\_SMS\_ACTION="SENT\_SMS\_ACTION"; 30. String DELIVERED\_SMS\_ACTION="DELIVERED\_SMS\_ACTION"; 31. SmsManager sms = SmsManager.getDefault(); 32. Intent sentIntent = new Intent("SENT\_SMS\_ACTION"); 33. PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,sentIntent, PendingIntent.FLAG\_IMMUTABLE); 34. Intent deliverIntent = new Intent("DELIVERED\_SMS\_ACTION"); 35. PendingIntent deliverPI = PendingIntent.getBroadcast(this, 0,deliverIntent, PendingIntent.FLAG\_IMMUTABLE); 36. registerReceiver(new BroadcastReceiver() 37. { 38. public void onReceive(Context \_context,Intent \_intent) 39. { 40. switch(getResultCode()){ 41. case AppCompatActivity.RESULT\_OK: 42. Toast.makeText(getBaseContext(),"短信发送成功", 43. Toast.LENGTH\_SHORT).show(); 44. break; 45. case SmsManager.RESULT\_ERROR\_GENERIC\_FAILURE: 46. Toast.makeText(getBaseContext(),"SMS generic failure actions", 47. Toast.LENGTH\_SHORT).show(); 48. break; 49. case SmsManager.RESULT\_ERROR\_RADIO\_OFF: 50. Toast.makeText(getBaseContext(),"SMS radio off failure actions", 51. Toast.LENGTH\_SHORT).show(); 52. break; 53. case SmsManager.RESULT\_ERROR\_NULL\_PDU: 54. Toast.makeText(getBaseContext(),"SMS null PDU failure actions", 55. Toast.LENGTH\_SHORT).show(); 56. break; 57. } 58. } 59. },new IntentFilter(SENT\_SMS\_ACTION), Context.RECEIVER\_NOT\_EXPORTED); 60. registerReceiver(new BroadcastReceiver() 61. { 62. public void onReceive(Context \_context, Intent \_intent) { 63. Toast.makeText(getBaseContext(), "SMS delivered actions", 64. Toast.LENGTH\_SHORT).show(); 65. } 66. }, new IntentFilter(DELIVERED\_SMS\_ACTION), Context.RECEIVER\_NOT\_EXPORTED); 67. if (texts.length() > 70) { 68. ArrayList<String> msgs = sms.divideMessage(texts); 69. for (String ms : msgs) { 70. sms.sendTextMessage(str\_num, null, ms, sentPI, deliverPI); 71. } 72. } else { 73. sms.sendTextMessage(str\_num, null, texts, sentPI, deliverPI); 74. } 75. Toast.makeText(getBaseContext(), "短信发送完成", Toast.LENGTH\_LONG).show(); 76. } 77. }   activity\_main.xml:   1. <?xml version="1.0" encoding="utf-8"?> 2. <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" 3. xmlns:app="http://schemas.android.com/apk/res-auto" 4. xmlns:tools="http://schemas.android.com/tools" 5. android:id="@+id/main" 6. android:layout\_width="match\_parent" 7. android:layout\_height="match\_parent" 8. tools:context=".MainActivity"> 9. <EditText 10. android:id="@+id/phone\_number" 11. android:layout\_width="0dp" 12. android:layout\_height="wrap\_content" 13. android:hint="请输入电话号码" 14. app:layout\_constraintEnd\_toEndOf="parent" 15. app:layout\_constraintStart\_toStartOf="parent" 16. app:layout\_constraintTop\_toTopOf="parent" /> 17. <EditText 18. android:id="@+id/input\_text" 19. android:layout\_width="0dp" 20. android:layout\_height="wrap\_content" 21. android:hint="请输入信息" 22. app:layout\_constraintEnd\_toStartOf="@id/send" 23. app:layout\_constraintStart\_toStartOf="parent" 24. app:layout\_constraintTop\_toBottomOf="@id/phone\_number" /> 25. <Button 26. android:id="@+id/send" 27. android:layout\_width="wrap\_content" 28. android:layout\_height="wrap\_content" 29. android:text="发送" 30. app:layout\_constraintEnd\_toEndOf="parent" 31. app:layout\_constraintTop\_toBottomOf="@id/phone\_number" /> 32. </androidx.constraintlayout.widget.ConstraintLayout>   **【实验结果】**    **【实验心得体会】**  通过本次实验，了解了短信控制的运行机制，熟悉了短信的发送方法，通过跟踪短信发送后的各个状态，对安卓提供的短信服务和如何编写短信相关程序有了更深入的理解。 | |

|  |  |
| --- | --- |
| **【实验编号】** | **3** |
| **【实验名称】** | **访问数据库，实现数据操作 学时：2** |
| **【实验目的】**  1）掌握SQLite嵌入式数据库使用。  2）掌握SQLite嵌入式数据库与Android Studio的连接与数据操作。  **【实验内容】**  建立一个数据库，在程序中访问数据库，实现数据增删改查功能  **【实验代码】**  SQLiteDemo:   1. public class SQLiteDemo extends AppCompatActivity { 2. private final MyDBHelper helper = new MyDBHelper(this); 3. protected void onCreate(Bundle savedInstanceState) { 4. super.onCreate(savedInstanceState); 5. EdgeToEdge.enable(this); 6. setContentView(R.layout.sqlite\_demo); 7. ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> { 8. Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars()); 9. v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom); 10. return insets; 11. }); 12. } 13. public void insert(View view) { 14. SQLiteDatabase db = helper.getWritableDatabase(); 15. ContentValues values = new ContentValues(); 16. AlertDialog.Builder builder = new AlertDialog.Builder(this); 17. EditText idTest = (EditText) this.findViewById(R.id.editText2); 18. EditText nameTest = (EditText) this.findViewById(R.id.editText1); 19. String id = idTest.getText().toString(); 20. String name = nameTest.getText().toString(); 21. String[] projection = { 22. "id", 23. "name" 24. }; 25. Cursor c = db.query( 26. "my\_table", 27. projection, 28. "id = ?", 29. new String[]{id}, 30. null, 31. null, 32. null 33. ); 34. if (c.moveToFirst()) { 35. db.close(); 36. builder.setTitle("提示"); 37. builder.setMessage("id已存在"); 38. builder.setPositiveButton("确定", null); 39. builder.show(); 40. return; 41. } 42. values.put("id", id); 43. values.put("name", name); 44. long newRowId = db.insert("my\_table", null, values); 45. db.close(); 46. builder.setTitle("提示"); 47. builder.setMessage("插入成功"); 48. builder.setPositiveButton("确定", null); 49. builder.show(); 50. } 51. public void query(View view) { 52. SQLiteDatabase db = helper.getReadableDatabase(); 53. AlertDialog.Builder builder = new AlertDialog.Builder(this); 54. String[] projection = { 55. "id", 56. "name" 57. }; 58. EditText idTest = (EditText) this.findViewById(R.id.editText2); 59. EditText nameTest = (EditText) this.findViewById(R.id.editText1); 60. String id = idTest.getText().toString(); 61. String name = nameTest.getText().toString(); 62. Cursor c = db.query( 63. "my\_table", 64. projection, 65. "id = ?", 66. new String[]{id}, 67. null, 68. null, 69. null 70. ); 71. if (c.moveToFirst()) { 72. int nameColumn = c.getColumnIndex("name"); 73. String n = c.getString(nameColumn); 74. nameTest.setText(n); 75. db.close(); 76. return; 77. } 78. db.close(); 79. builder.setTitle("提示"); 80. builder.setMessage("id不存在"); 81. builder.setPositiveButton("确定", null); 82. builder.show(); 83. } 84. }   MyDBHelper:   1. public class MyDBHelper extends SQLiteOpenHelper{ 2. private static final String DATABASE\_NAME = "my.db"; 3. private static final int DATABASE\_VERSION = 1; 4. public MyDBHelper(Context context) { 5. super(context, DATABASE\_NAME, null, DATABASE\_VERSION); 6. } 7. public void onCreate(SQLiteDatabase db) { 8. String createSQL = "CREATE TABLE"+" my\_table"+ "("+"id "+"INTEGER PRIMARY KEY,"+"name "+"TEXT"+")"; 9. db.execSQL(createSQL); 10. } 11. @Override 12. public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) { 13. } 14. }   sqlite\_demo:   1. <?xml version="1.0" encoding="utf-8"?> 2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" 3. xmlns:tools="http://schemas.android.com/tools" 4. android:layout\_width="match\_parent" 5. android:layout\_height="match\_parent" 6. android:id="@+id/main" 7. tools:context=".SQLiteDemo"> 8. <TextView 9. android:id="@+id/textView1" 10. android:layout\_width="wrap\_content" 11. android:layout\_height="wrap\_content" 12. android:layout\_alignParentLeft="true" 13. android:layout\_alignParentTop="true" 14. android:layout\_marginLeft="44dp" 15. android:layout\_marginTop="57dp" 16. android:text="编号：" /> 17. <TextView 18. android:id="@+id/textView2" 19. android:layout\_width="wrap\_content" 20. android:layout\_height="wrap\_content" 21. android:layout\_alignLeft="@+id/textView1" 22. android:layout\_below="@+id/textView1" 23. android:layout\_marginTop="88dp" 24. android:text="名称：" /> 25. <EditText 26. android:id="@+id/editText1" 27. android:layout\_width="wrap\_content" 28. android:layout\_height="wrap\_content" 29. android:layout\_alignLeft="@+id/textView2" 30. android:layout\_centerVertical="true" 31. android:ems="10"> 32. <requestFocus /> 33. </EditText> 34. <EditText 35. android:id="@+id/editText2" 36. android:layout\_width="wrap\_content" 37. android:layout\_height="wrap\_content" 38. android:layout\_alignStart="@+id/textView1" 39. android:layout\_below="@+id/textView1" 40. android:layout\_marginTop="20dp" 41. android:ems="10" 42. android:inputType="number" /> 43. <Button 44. android:id="@+id/button1" 45. android:layout\_width="wrap\_content" 46. android:layout\_height="wrap\_content" 47. android:layout\_alignStart="@+id/editText1" 48. android:layout\_below="@+id/editText1" 49. android:layout\_marginTop="73dp" 50. android:text="插入" 51. android:onClick="insert"/> 52. <Button 53. android:id="@+id/button2" 54. android:layout\_width="wrap\_content" 55. android:layout\_height="wrap\_content" 56. android:layout\_alignBottom="@+id/button1" 57. android:layout\_alignRight="@+id/editText1" 58. android:layout\_marginRight="22dp" 59. android:text="查询" 60. android:onClick="query"/> 61. </RelativeLayout>   **【实验结果】**      **【实验心得体会】**  通过本次实验，掌握了SQLite嵌入式数据库的使用,掌握了SQLite嵌入式数据库与Android Studio的连接与数据操作，使用SQLite编写了简单的增删改查程序，为下一步编写持久化的安卓程序打下基础。 | |

|  |  |
| --- | --- |
| **【实验编号】** | **4** |
| **【实验名称】** | **广播的发送与过滤机制 学时：2** |
| **【实验目的】**  1）了解广播的相关概念及运行机制  2）熟悉广播接收者的过滤机制  **【实验内容】**  设计一个简单的广播程序，采用Notification Manager来提示广播信息，然后再运行程序，查看运行结果，实现广播的提醒与过滤功能。  **【实验代码】**  BroadcastActivity.java:   1. public class BroadcastActivity extends AppCompatActivity { 2. protected void onCreate(Bundle savedInstanceState) { 3. super.onCreate(savedInstanceState); 4. EdgeToEdge.enable(this); 5. setContentView(R.layout.broadcast\_send); 6. ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> { 7. Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars()); 8. v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom); 9. return insets; 10. }); 11. LocalBroadcastReceiver receiver = new LocalBroadcastReceiver(); 12. IntentFilter intentFilter = new IntentFilter(); 13. intentFilter.addAction("abc3"); 14. registerReceiver(receiver, intentFilter, Context.RECEIVER\_NOT\_EXPORTED); 15. Button btn = findViewById(R.id.BCbutton1); 16. btn.setOnClickListener(new View.OnClickListener() { 17. @Override 18. public void onClick(View v) { 19. Intent intent = new Intent(); 20. intent.setAction("abc3"); 21. sendBroadcast(intent); 22. } 23. }); 24. } 25. class LocalBroadcastReceiver extends BroadcastReceiver { 26. private NotificationManager manager; 27. LocalBroadcastReceiver() { 28. super(); 29. } 30. @Override 31. public void onReceive(Context context, Intent intent) { 32. if (intent.getAction().equals("abc1")) { 33. Toast.makeText(context, "广播来了1：", 1).show(); 34. } 35. if (intent.getAction().equals("abc2")) { 36. Toast.makeText(context, "广播来了2：", 1).show(); 37. } 38. if (intent.getAction().equals("abc3")) { 39. manager = (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE); 40. if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) { 41. NotificationChannel channel = new NotificationChannel("channelid1", "channelname", NotificationManager.IMPORTANCE\_HIGH); 42. channel.setDescription("这是channel1"); 43. channel.enableLights(true); 44. channel.enableVibration(true); 45. channel.setLightColor(Color.RED); 46. manager.createNotificationChannel(channel); 47. } 48. NotificationCompat.Builder builder = new NotificationCompat.Builder(context, "channelid1"); 49. builder.setSmallIcon(R.drawable.ic\_launcher\_background) 50. .setContentTitle("没有草莓") 51. .setContentText("注意米线！") 52. .setAutoCancel(true); 53. manager.notify(0x12, builder.build()); 54. } 55. } 56. } 57. }   BroadcastActivity.java:   1. <?xml version="1.0" encoding="utf-8"?> 2. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" 3. xmlns:tools="http://schemas.android.com/tools" 4. android:layout\_width="match\_parent" 5. android:layout\_height="match\_parent" 6. android:id="@+id/main" 7. tools:context=".BroadcastActivity"> 8. <TextView 9. android:layout\_width="wrap\_content" 10. android:layout\_height="wrap\_content" 11. android:text="广播发送程序" /> 12. <Button 13. android:id="@+id/BCbutton1" 14. android:layout\_width="wrap\_content" 15. android:layout\_height="wrap\_content" 16. android:layout\_centerHorizontal="true" 17. android:layout\_centerVertical="true" 18. android:text="发送广播" /> 19. </RelativeLayout>   **【实验结果】**    **【实验心得体会】**  通过本次实验，了解了广播的相关概念及运行机制，熟悉了广播接收者的过滤机制，编写了简单的广播程序，采用Notification Manager来提示广播信息，冰城实现广播的提醒与过滤功能，对广播有了更深入的认识与理解。 | |