Android 笔试题

一、选择题

1. **String s = null;** 会抛出异常的是**( )(多选)**
2. if((s != null) & (s.length() > 0))
3. if((s != null) && (s.length() > 0))
4. if((s == null) | (s.length() == 0))
5. if((s == null) || (s.length() == 0))
6. **swithch(){}** 结构中，括号中可使用的数据类型不能是**( )(**多选**)**

A. long B. Int C. short D.float

1. **SharedPreferences** 不支持存放哪种数据类型**( )**

A. boolean B. int C. String D. double

1. 以下哪个不是 **Activity** 启动时的正确标志**( )**
2. FLAG\_ACTIVITy\_CLEAT\_TOP B. FLAG\_ACTIVITY\_SINGLE\_TASK

C. FLAG\_ACTIVITy\_SINGLE\_TOP D. FLAG\_ACTIVITY\_NEW\_TASK

1. 关于 **ServiceConnection** 接口的 **onServiceConnected** 方法的触发条件描述正确的是**( )**
2. bindService()方法执行成功后
3. bindService()方法执行成功同时 onBind()返回非空IBinder对象
4. Service 的 onCreate()方法和onBind()方法执行成功后
5. Service 的 onCreate()方法和onStartCommand()方法启动成功后
6. 下面关于谁先接收广播顺序错误的是**( )**
7. 有序广播，优先级高的先接收。
8. 有序广播，同优先级的动静态广播接收器，静态优先于动态。
9. 有序广播，同优先级的动态广播接收器，先注册的大于后注册的。
10. 普通广播，无视优先级，动态广播接收者优先于静态广播接收器
11. 下面哪条语句可以构造正确的对话框**( )**
12. AlertDialog dialog = new AlertDialog(context);
13. AlertDialog.Builder builder = new AlertDialog.Builder(context);
14. ProgressDialog dialog = new ProgressDialog();
15. ProgressDialog.Builder builder = new ProgressDialog.Builder(context);
16. 在**Android**中使用**Menu**时可能需要重写的方法有**( )(**多选**)**
17. onCreateOptionsMenu()
18. onCreateMenu()
19. onOptionsItemSelected()
20. onItemSelected()
21. **Android** 提供了哪个高效序列化接口 **( )**

A. ContentResolver B. Parcelable C. Copiable D. Externalizable

二、编程题

1. 请使用Java 语言 写出一种单例模式的实现方式

|  |
| --- |
|  |

1. 请使用Java 语言 写出一种 冒泡排序 算法的实现方式

|  |
| --- |
|  |

1. 请指出下面这段代码的设计缺陷

|  |
| --- |
| public class MainActivity extends AppCompatActivity {  @BindView(R.id.tv\_time)  TextView tvTime;  private int time = 6000;  private Handler mHandle;  @Override  protected void onCreate(@Nullable Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  }  @OnClick(R.id.bt\_start)  void start() {  mHandle.post(runnable);  }  Runnable runnable = new Runnable() {  @Override  public void run() {  time--;  tvTime.setText(time);  mHandle.postDelayed(runnable, 1000);  }  };  @Override  protected void onDestroy() {  super.onDestroy();  }  } |