



Englisch



Deutsch ▼



Captainandroid



Ultimate Room DB-Tutorial: Installieren Sie Room DB in der Android-App



TUSHAR / 23. APRIL 2022 / ANDROID , STARTSEITE , TUTORIALS , NICHT KATEGORISIERT

Beitragsaufrufe: 194

Wenn Sie Daten lokal in Ihrer App speichern müssen, ist die Raumdatenbank das beste Datenbanksystem für Android. Heute werden wir das Raum-DB-Tutorial beschreiben und lernen, wie man die Raumdatenbank in Android mit Java implementiert und verwendet.

Inhaltsverzeichnis



1. Schritt 1: Room DB installieren
2. Schritt 2: Tabellen erstellen
3. Schritt 2: DAO erstellen
4. Schritt 3: AppDatabase erstellen
5. Schritt 4: DbClient erstellen
6. Schritt 5: App ausführen und testen

Schritt 1: Room DB installieren

Öffnen Sie zunächst **build.gradle** auf Projektebene . Fügen Sie den folgenden Code in die **Abhängigkeiten** und den **defaultConfig**- Block von Android ein .

```
dependencies {
    ....
    ....
    //room database
    implementation 'androidx.room:room-runtime:2.4.1'
    annotationProcessor 'androidx.room:room-compiler:2.4.1'
}

android {
    ....
    ....

    defaultConfig {
        ....
        ....

        javaCompileOptions {
            annotationProcessorOptions {
                arguments = ["room.schemaLocation": "$projectDir/
            }
        }
    }

    sourceSets {
        androidTest.assets.srcDirs += files("$projectDir/schemas
    }
}
```

Ihre Schemadatei Ihrer Datenbankversion wird im Ordner „schemas“ gespeichert.

Schritt 2: Tabellen erstellen

Erstellen wir ein neues Paket und nennen es **db**. In diesem Paket erstellen wir alle datenbankbezogenen Dateien.

Erstellen Sie nun ein weiteres Paket in **db**. Nennen Sie es **Tische**. Nehmen wir an, wir möchten eine Tabelle mit Büchern erstellen. In der Buchtabelle möchten wir die Spalten

„ID“, „Name“, „Autor“, „PublishDate“, „CreatedAt“ und „UpdatedAt“ haben .

Erstellen Sie also im Tabellenpaket eine Klasse mit dem Namen **Book** . Fügen Sie den folgenden Code ein:

```
import androidx.room.PrimaryKey;
import androidx.room.Entity;

@Entity
public class Book {
    @PrimaryKey(autoGenerate = true)
    private int id;
    private String name;
    private String author;
    private long publishDate;
    private long createdAt;
    private long updatedAt;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public long getPublishDate() {
```

```
        return publishDate;
    }

    public void setPublishDate(long publishDate) {
        this.publishDate = publishDate;
    }

    public long getCreatedAt() {
        return createdAt;
    }

    public void setCreatedAt(long createdAt) {
        this.createdAt = createdAt;
    }

    public long getUpdatedAt() {
        return updatedAt;
    }

    public void setUpdatedAt(long updatedAt) {
        this.updatedAt = updatedAt;
    }
}
```

Hier bedeutet **@Entity** , dass wir diese Klasse als Tabelle und **@PrimaryKey(autoGenerate = true)** deklarieren, um die Feld- **ID** als Primärschlüssel und automatische Inkrementierung festzulegen .

Schritt 2: DAO erstellen

Erstellen Sie auf ähnliche Weise ein weiteres Paket in **db** und nennen Sie es **dao** (Data Access Object). Wir werden das ganze **Chaos** hier behalten. Erstellen Sie als Nächstes eine Schnittstelle mit dem Namen **BookDao** und fügen Sie den folgenden Code ein:

```

import androidx.room.Dao;
import androidx.room.Insert;
import androidx.room.Query;
import androidx.room.Update;

import com.captainandroid.roomdbtutorial.db.tables.Book;

import java.util.List;

@Dao
public interface BookDao {
    @Insert
    long insertBook(Book book);

    @Update
    int updateBook(Book book);

    @Query("Select * from book")
    List<Book> getAllBooks();
}

```

@Dao- Annotation, die zum Deklarieren einer Schnittstelle als DAO verwendet wird.

[Einzelheiten zu allen Anmerkungen können Sie hier](#) nachlesen .

Schritt 3: AppDatabase erstellen

Erstellen Sie anschließend im **Datenbankpaket** eine weitere Klasse und nennen Sie sie **AppDatabase** . Fügen Sie den folgenden Code ein:

```

import androidx.room.Database;
import androidx.room.RoomDatabase;

import com.captainandroid.roomdbtutorial.db.dao.BookDao;
import com.captainandroid.roomdbtutorial.db.tables.Book;

@Database(entities = {Book.class}, exportSchema = true, version = 1)
public abstract class AppDatabase extends RoomDatabase {

```

```
public abstract BookDao getBookDao();  
}
```

Schritt 4: DbClient erstellen

Erstellen Sie auf ähnliche Weise eine weitere Klasse im **Datenbankpaket** und nennen Sie sie **DbClient**. Fügen Sie den folgenden Code ein:

```
import android.content.Context;  
  
import androidx.room.Room;  
  
public class DbClient {  
    private Context mContext;  
    private static DbClient mInstance;  
  
    private AppDatabase appDatabase;  
  
    private DbClient(Context mContext) {  
        this.mContext = mContext;  
        appDatabase = Room.databaseBuilder(mContext, AppDatabase.class,  
            "allowMainThreadQueries()  
            build();  
    }  
  
    public static synchronized DbClient getInstance(Context mContext)  
    {  
        if (mInstance == null) {  
            mInstance = new DbClient(mContext);  
        }  
        return mInstance;  
    }  
  
    public AppDatabase getAppDatabase() {  
        return appDatabase;  
    }  
}
```

Hier ist die Paket- und Klassenstruktur:

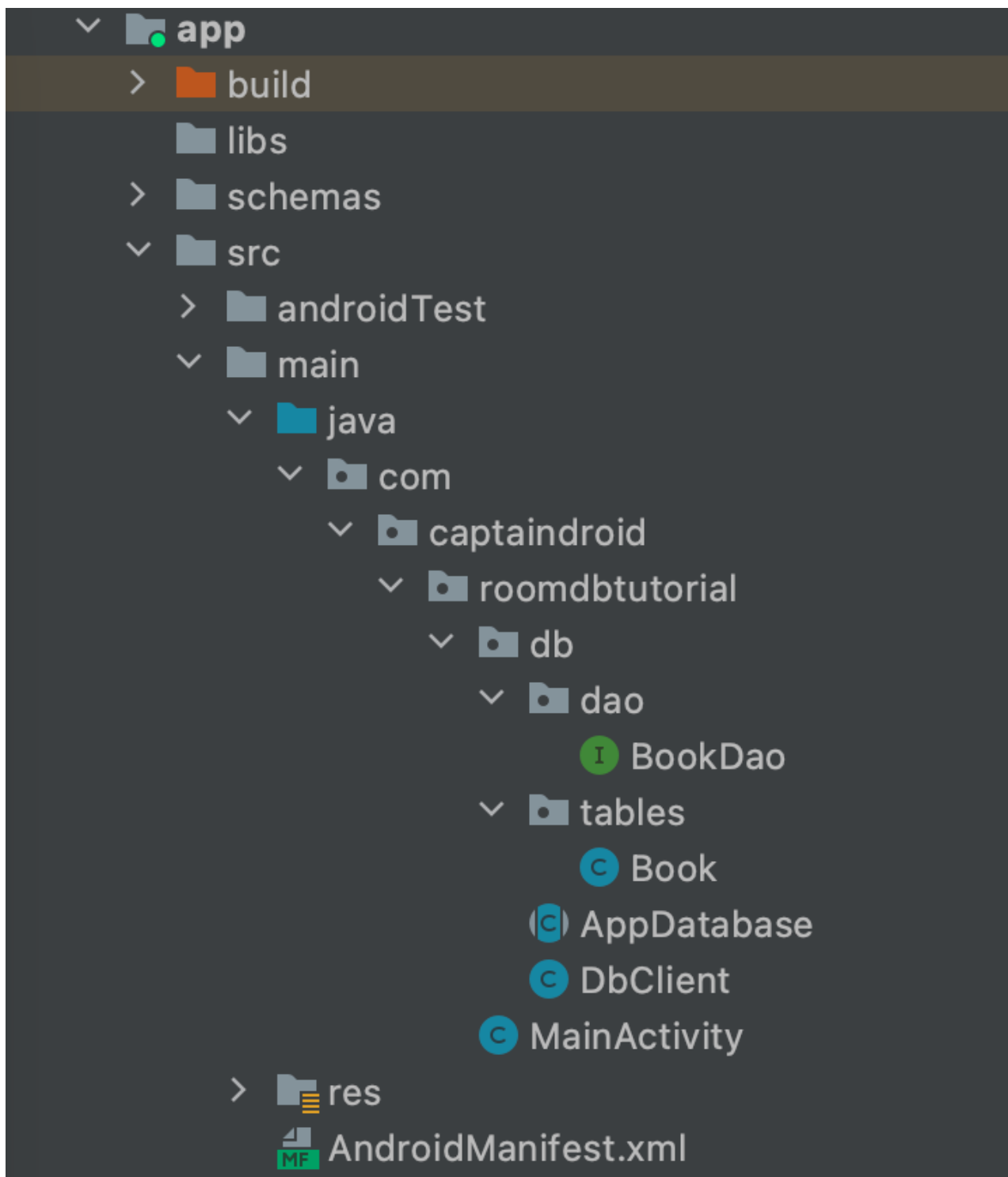


Bild der Paketstruktur.

Schritt 5: App ausführen und testen

Darüber hinaus finden Sie hier Codes, die die Nutzung des gesamten Projekts beschreiben. Sie können die App jetzt beispielsweise mit den folgenden Codes ausführen und testen.

Hauptaktivität :

```
package com.captainandroid.roomdbtutorial;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.captainandroid.roomdbtutorial.databinding.ActivityMainB
import com.captainandroid.roomdbtutorial.databinding.RowBookBindin
import com.captainandroid.roomdbtutorial.db.DbClient;
import com.captainandroid.roomdbtutorial.db.tables.Book;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding;

    private List<Book> books = new ArrayList<>();
    private Adapter adapter;
    private DbClient dbClient;

    private SimpleDateFormat simpleDateFormat = new SimpleDateFo
```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityMainBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    dbClient = DbClient.getInstance(this);
    adapter = new Adapter();

    books.addAll(dbClient.getAppDatabase().getBookDao().getAll());

    binding.rvBookList.setLayoutManager(new LinearLayoutManager(this));
    binding.rvBookList.setAdapter(adapter);

    binding.btnAddBook.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Book book = new Book();
            book.setName("Book " + System.currentTimeMillis());
            book.setAuthor("Author " + System.currentTimeMillis());
            book.setPublishDate(System.currentTimeMillis());
            book.setCreatedAt(System.currentTimeMillis());
            book.setUpdatedAt(System.currentTimeMillis());

            dbClient.getAppDatabase().getBookDao().insertBook(book);

            books.clear();
            books.addAll(dbClient.getAppDatabase().getBookDao().getAll());
            adapter.notifyDataSetChanged();
        }
    });
}

private class Adapter extends RecyclerView.Adapter<ViewHolder> {

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new ViewHolder(RowBookBinding.inflate(getLayoutInflater(), parent, false));
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        Book book = books.get(position);
        holder.binding.tvTitle.setText(book.getName());
        holder.binding.tvAuthor.setText(book.getAuthor());
        holder.binding.tvPublishDate.setText(book.getPublishDate());
        holder.binding.tvCreatedAt.setText(book.getCreatedAt());
        holder.binding.tvUpdatedAt.setText(book.getUpdatedAt());
    }

    @Override
    public int getItemCount() {
        return books.size();
    }
}

```

```

public void onBindViewHolder(@NonNull ViewHolder holder,
    holder.rowBookBinding.tvId.setText(books.get(position)
    holder.rowBookBinding.tvBookName.setText(books.get(position)
    holder.rowBookBinding.tvAuthorName.setText(books.get(position)
    holder.rowBookBinding.tvPublishDate.setText(books.get(position)

    if(position == books.size() - 1){
        holder.rowBookBinding.vLastDiv.setVisibility(View.VISIBLE)
    }else {
        holder.rowBookBinding.vLastDiv.setVisibility(View.INVISIBLE)
    }
}

@Override
public int getItemCount() {
    return books.size();
}
}

private class ViewHolder extends RecyclerView.ViewHolder{
    RowBookBinding rowBookBinding;
    public ViewHolder(RowBookBinding rowBookBinding) {
        super(rowBookBinding.getRoot());
        this.rowBookBinding = rowBookBinding;
    }
}
}

```

Layout :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:background="#858585"

```

```
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:paddingLeft="15dp"
            android:textColor="@color/black"
            android:layout_weight="0.4"
            android:text="Id"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:layout_weight="1"
            android:text="Name"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:layout_weight="1"
            android:text="Author"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:layout_weight="1"
            android:text="Publish Date"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>
    </LinearLayout>

    <LinearLayout
        android:layout_weight="1"
        android:padding="15dp"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rvBookList"
            android:layout_weight="1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>

        <Button
```

```

        android:id="@+id/btAddBook"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add Book"
        android:layout_gravity="bottom|center" />
    </LinearLayout>

```

```

</LinearLayout>

```

row_book:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <View
        android:background="@color/black"
        android:layout_width="match_parent"
        android:layout_height=".5dp"/>

    <LinearLayout
        android:gravity="center"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="30dp">
        <View
            android:layout_marginRight="2dp"
            android:background="@color/black"
            android:layout_width="0.5dp"
            android:layout_height="match_parent"/>

        <TextView
            android:gravity="center"
            android:id="@+id/tvId"
            android:textColor="@color/black"
            android:layout_weight="0.2"
            android:text="Id"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>

        <View
            android:layout_marginLeft="2dp"

```

```
        android:layout_marginRight="2dp"
        android:background="@color/black"
        android:layout_width="0.5dp"
        android:layout_height="match_parent"/>
<TextView
    android:singleLine="true"
    android:id="@+id/tvBookName"
    android:textColor="@color/black"
    android:layout_weight="1"
    android:text="Name"
    android:layout_width="0dp"
    android:layout_height="wrap_content"/>
<View
    android:layout_marginLeft="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/black"
    android:layout_width="0.5dp"
    android:layout_height="match_parent"/>
<TextView
    android:singleLine="true"
    android:id="@+id/tvAuthorName"
    android:textColor="@color/black"
    android:layout_weight="1"
    android:text="Author"
    android:layout_width="0dp"
    android:layout_height="wrap_content"/>
<View
    android:layout_marginLeft="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/black"
    android:layout_width="0.5dp"
    android:layout_height="match_parent"/>
<TextView
    android:singleLine="true"
    android:id="@+id/tvPublishDate"
    android:textColor="@color/black"
    android:layout_weight="1"
    android:text="Publish Date"
    android:layout_width="0dp"
    android:layout_height="wrap_content"/>
<View
    android:layout_marginLeft="2dp"
    android:background="@color/black"
```

```
        android:layout_width="0.5dp"
        android:layout_height="match_parent"/>
</LinearLayout>

<View
    android:id="@+id/vLastDiv"
    android:background="@color/black"
    android:layout_width="match_parent"
    android:layout_height=".5dp"/>

</LinearLayout>
```

6:32



Room DB Tutorial

Id	Name	Author	Publish Date
----	------	--------	--------------

1	Book 16507159...	Author 1650715...	23 Sat 22, 06:12
2	Book 16507159...	Author 1650715...	23 Sat 22, 06:12
3	Book 16507159...	Author 1650715...	23 Sat 22, 06:13
4	Book 16507162...	Author 1650716...	23 Sat 22, 06:16
5	Book 16507171...	Author 1650717...	23 Sat 22, 06:32
6	Book 16507171...	Author 1650717...	23 Sat 22, 06:32

ADD BOOK

Aussehen der App. Auflistung aller Bücher aus der Datenbank.

Darüber hinaus finden Sie **im Github-Repository** auch:

Github-Repository



Android

Android Studio

Datenbank

#java

Raumdatenbank



VORHERIGER

NÄCHSTER

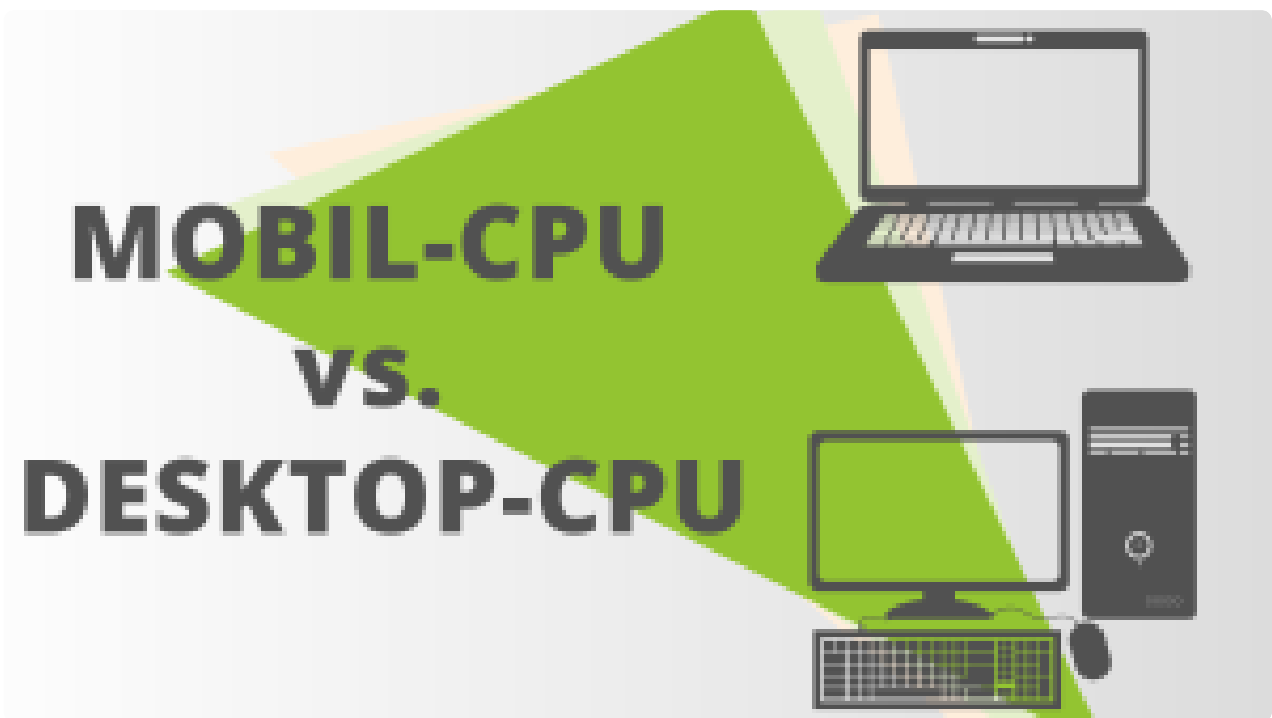


zusammenhängende Posts



Android-Betriebssystem im Detail

3. Januar 2023



Bemerkenswerter Vergleich: Laptop-Prozessor vs. Desktop-Prozessor

1. Januar 2023



Überwachen Sie die Leistung Ihrer Website auf diese überraschenden Vorteile


16. August 2022

ALSO ON CAPTAINANDROID.COM


<p>Remarkable compare: Laptop processor vs ...</p> <p>6 months ago</p> <p>Remarkable compare: Laptop processor vs desktop processor ...</p>	<p>Android operation system in details</p> <p>5 months ago</p> <p>Android operation system in details Android is a mobile operating system ...</p>	<p>Classes and Objects in Java</p> <p>4 months ago</p> <p>Classes and Objects in Java</p>
--	---	--

What do you think?


1 Response




Upvote




Funny




Love



Surprised



Angry



Sad

0 Comments

G

Start the discussion...

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name



Share

Best

Be the first to comment.

Subscribe Privacy Do Not Sell My Data