

Föreläsning 1

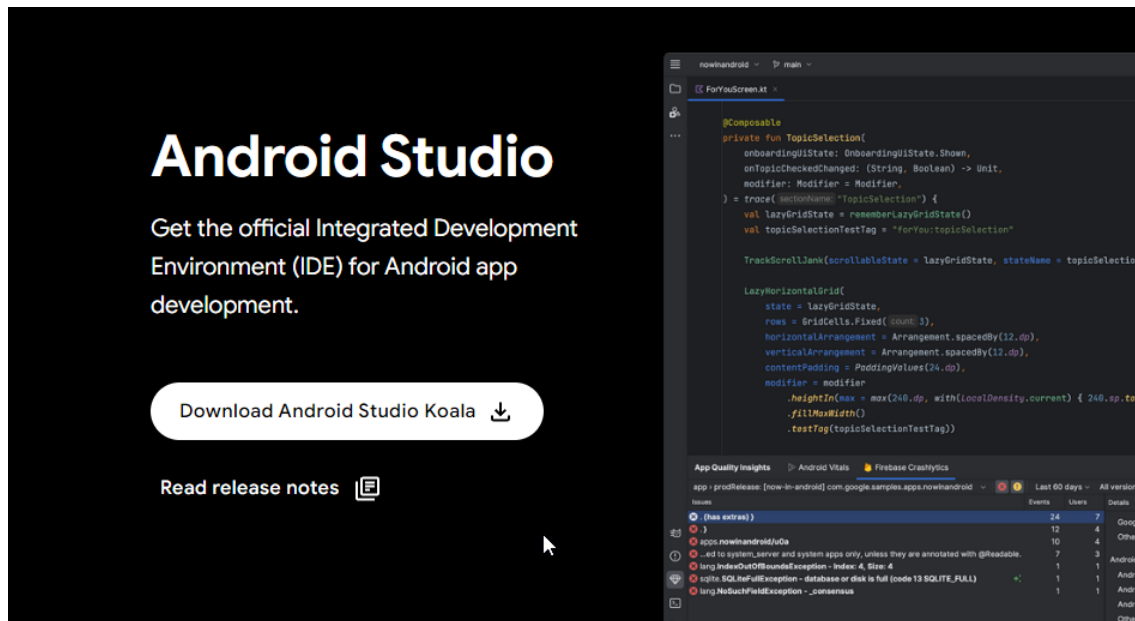


Innehåll

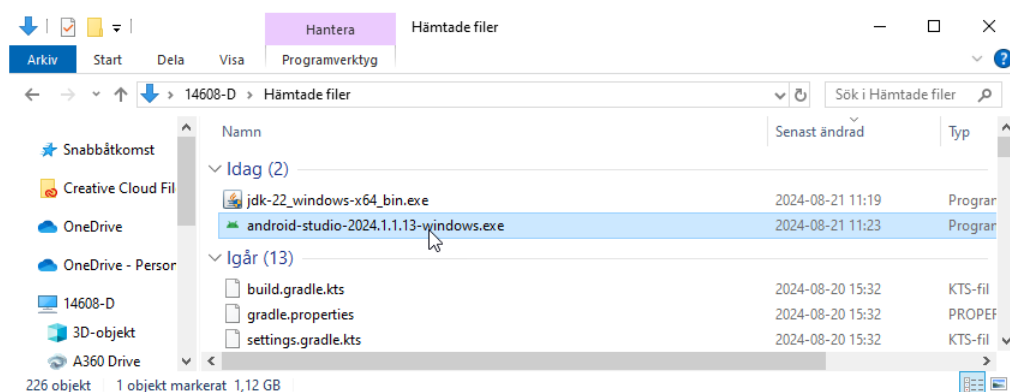
Kör din app på emulatorn	2
Använd verktygsfönster	10
Använd menyer och verktygslådor.....	10
Sök efter filer och kod	11
Skapa ny Kotlin fil	12

Introduktion till att skapa appar för android

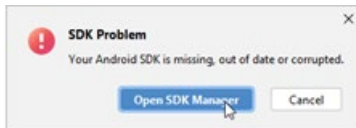
Vi vill använda den senaste versionen av Android Studio (AS) för att utveckla våra Android-appar. Android Studio är den officiella utvecklingsmiljön för Android. Den inkluderar verktyg för att designa, koda, testa och felsöka Android-appar. AS använder sig av Java och Gradle som bygger ditt projekt. Android Studio levereras med en integrerad JDK (Java Development Kit) och Gradle, så du behöver inte installera dessa separat.



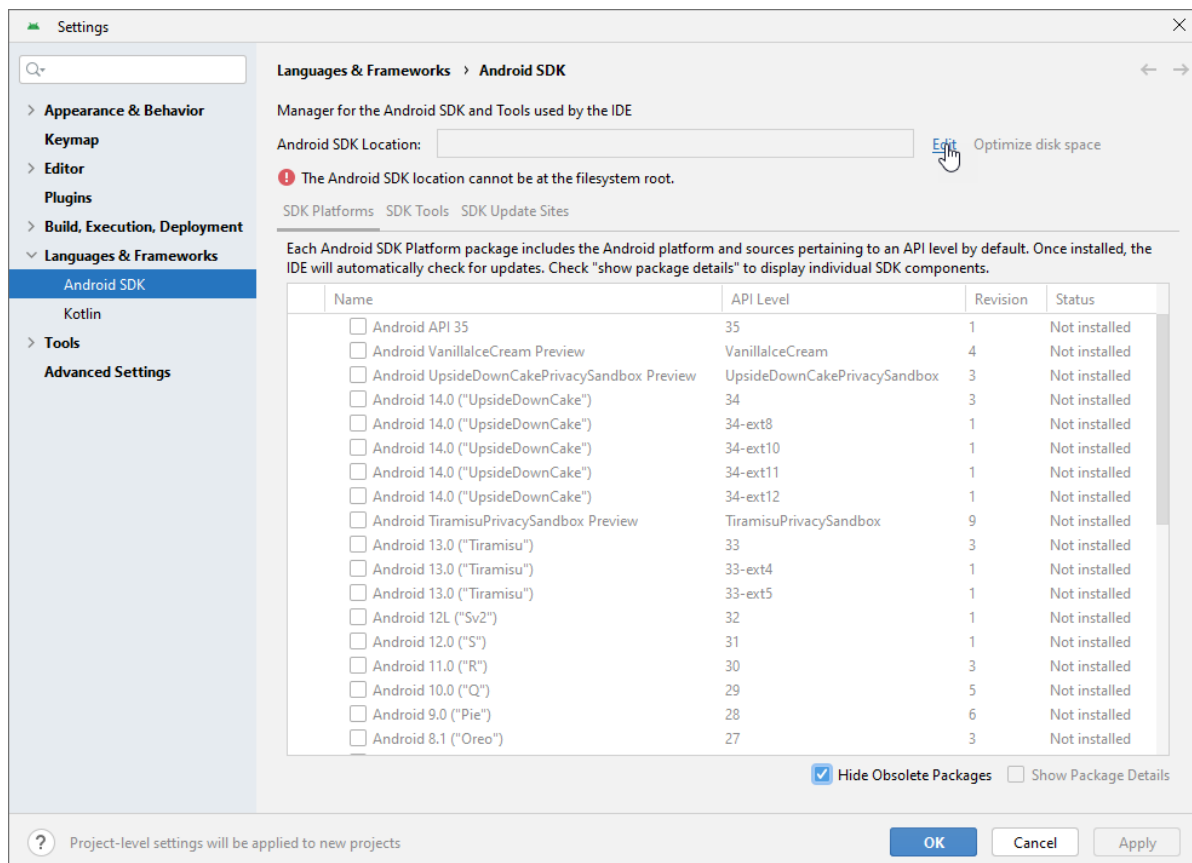
Kör den nedladdade installationsfilen genom att klicka på den. Följ installationsguiden och acceptera alla standardinställningar Genom att klicka på [Next](#) i varje steg. Detta säkerställer att Android Studio är korrekt installerad med standard konfigurationen.



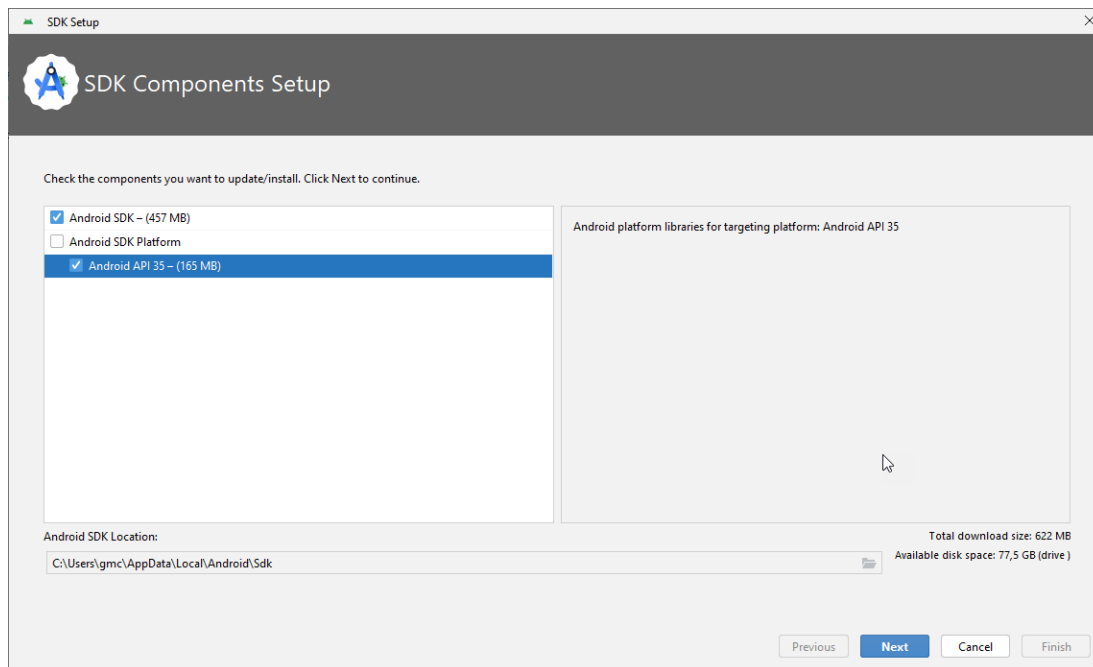
Android Studio använder olika SDK för olika android versioner. Vid första uppstart kan SDK för ditt projekt saknas, och du får antagligen ett felmeddelande när du försöker skapa ett nytt projekt. För att fixa detta, klicka på [Open SDK Manager](#) i felmeddelandet.



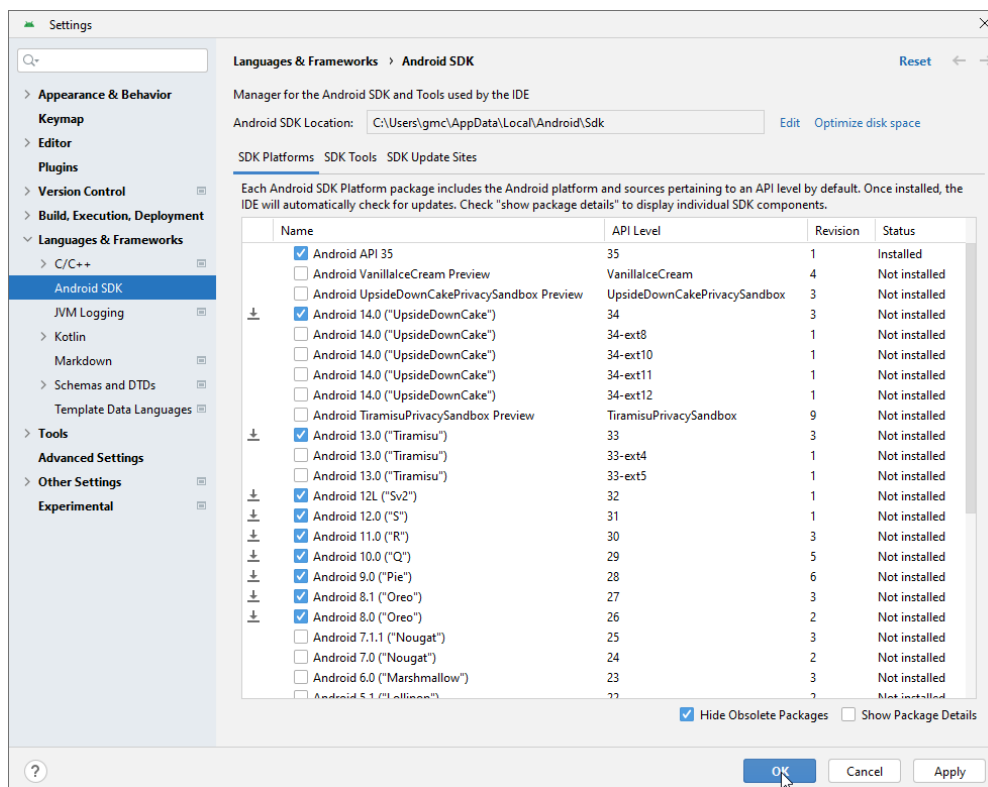
Du kommer nu till inställningar för språk och ramverk. Klicka på *Android SDK* till vänster, och sedan på *OK* för att öppna SDK inställningarna. Detta steg är nödvändigt för att välja och installera rätt SDK för ditt projekt.



Android SDK API 35 är förvald och rekommenderas för de senaste versionerna av Android. Klicka på *Next* tills installationen av SDK är klar.

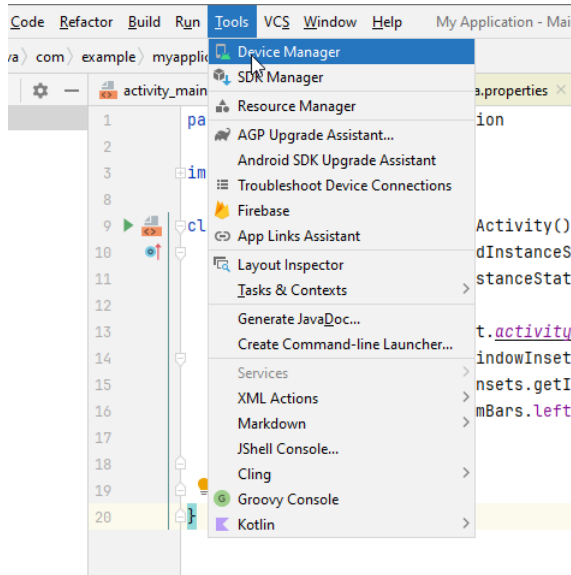


För att stödja äldre versioner av Android behöver vi installera SDK för dessa versioner också. Öppna inställningarna igen, *File > Settings > Languages & Frameworks > Android SDK*. Markera de paket som är markerade på bilden och klicka på Apply för att installera dem.

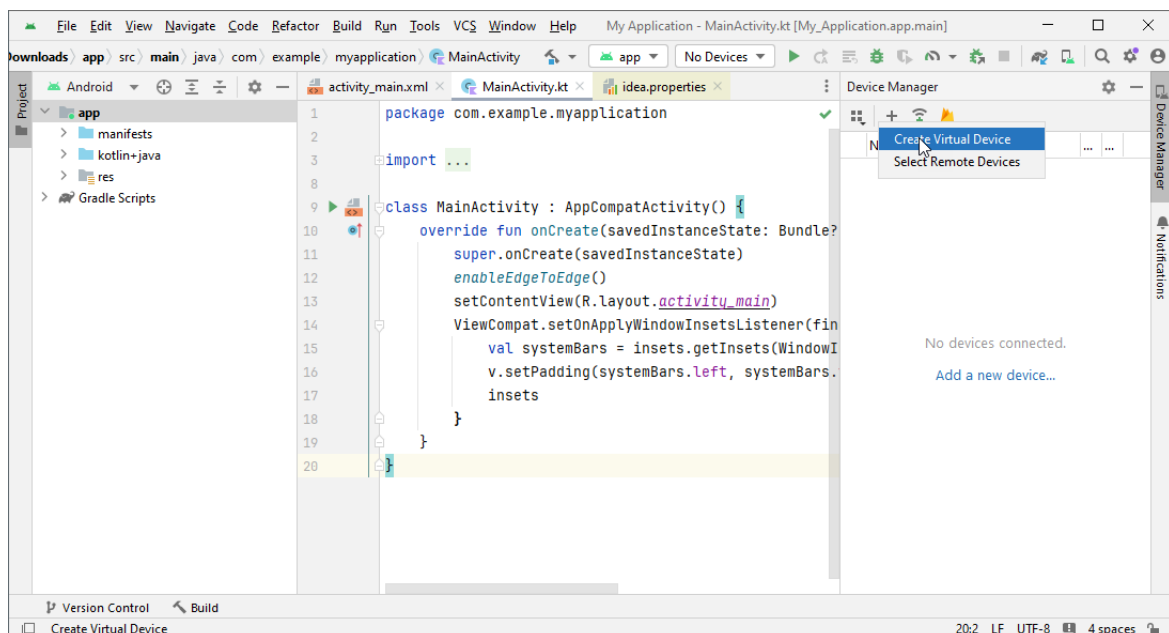


Introduktion till att skapa appar för android

Android Studio kommer med virtuella enheter, eller emulatorer, där du kan testa dina appar utan att behöva en fysisk enhet. Nu ska vi skapa en emulator för att testa våra appar.

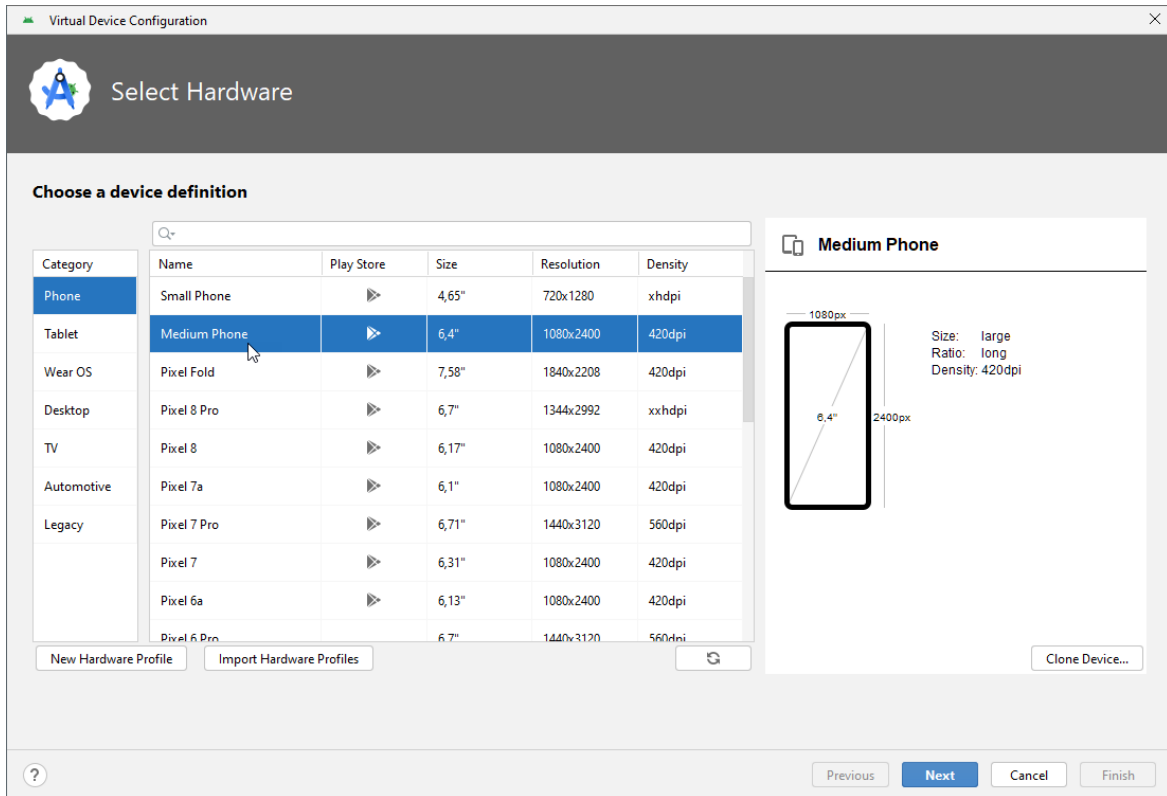


För att skapa en emulator, öppna *Tools > Device Manager*. Device Manager visas i den högra panelen. Klicka på "+" och välj *Create Virtual Device* för att starta processen.

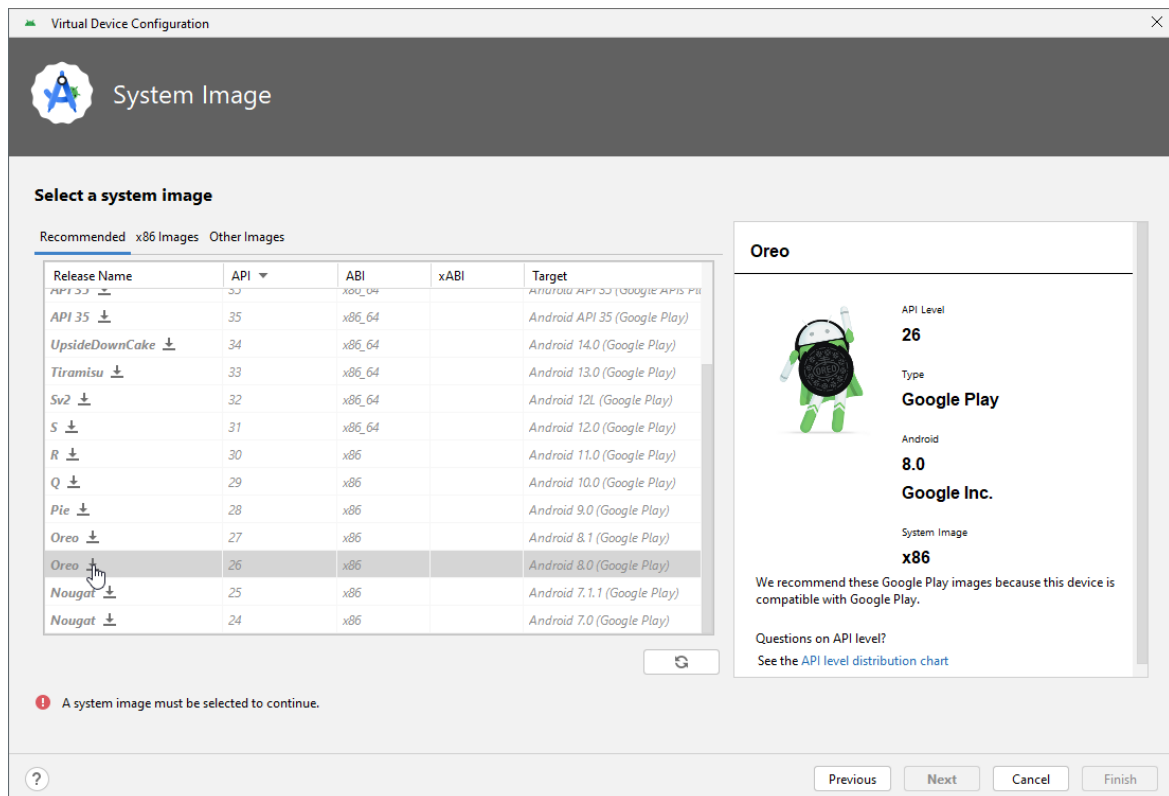


Introduktion till att skapa appar för android

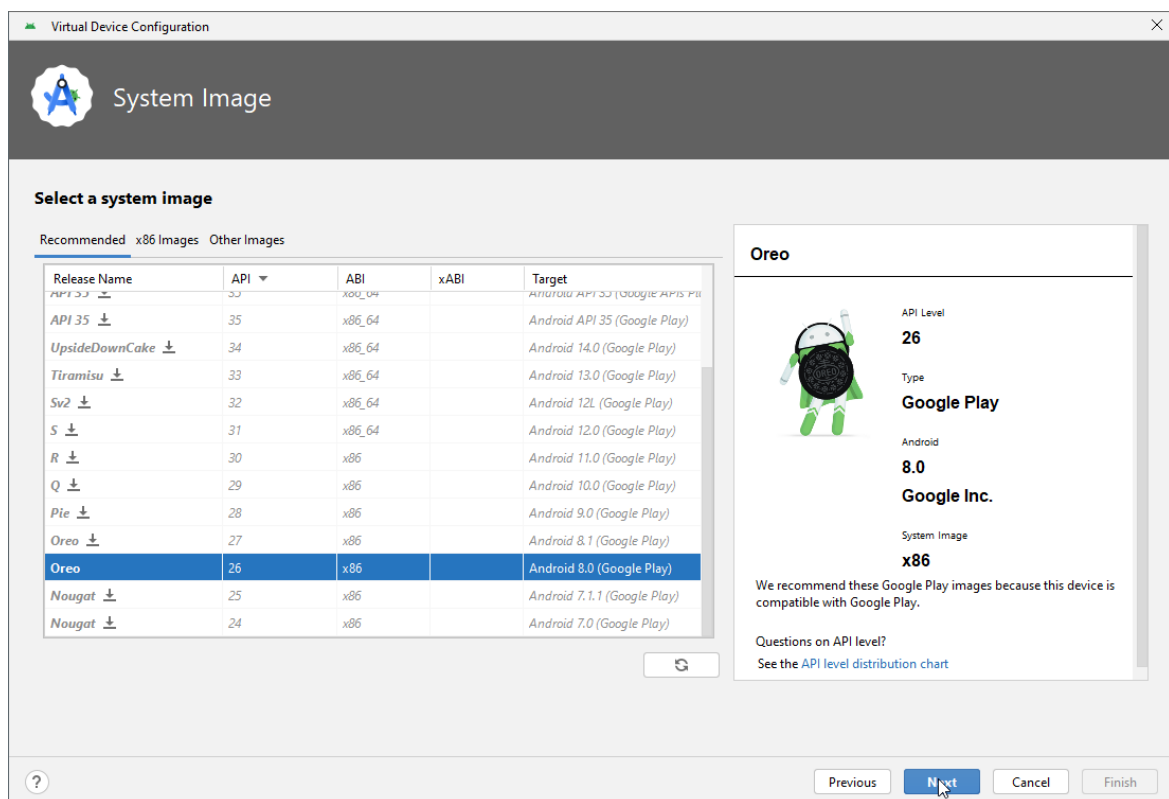
Välj en emulator från listan med tillgängliga enheter. I exemplet på bilden skapar jag en generisk emulator med medelstor skärm, men du kan välja den som passar dig bäst. Klicka på [Next](#) för att gå vidare till nästa steg.



Från början finns det ingen System Image installerad, vilket är nödvändigt för att köra emulator. Du behöver en system image för varje android version du vill utveckla för. Vi ska i kursen programmera för Android Oreo, så installera Oreo system image genom att klicka på download-ikonen bredvid. Klicka på [Next](#) i nästa fönster för att fortsätta.

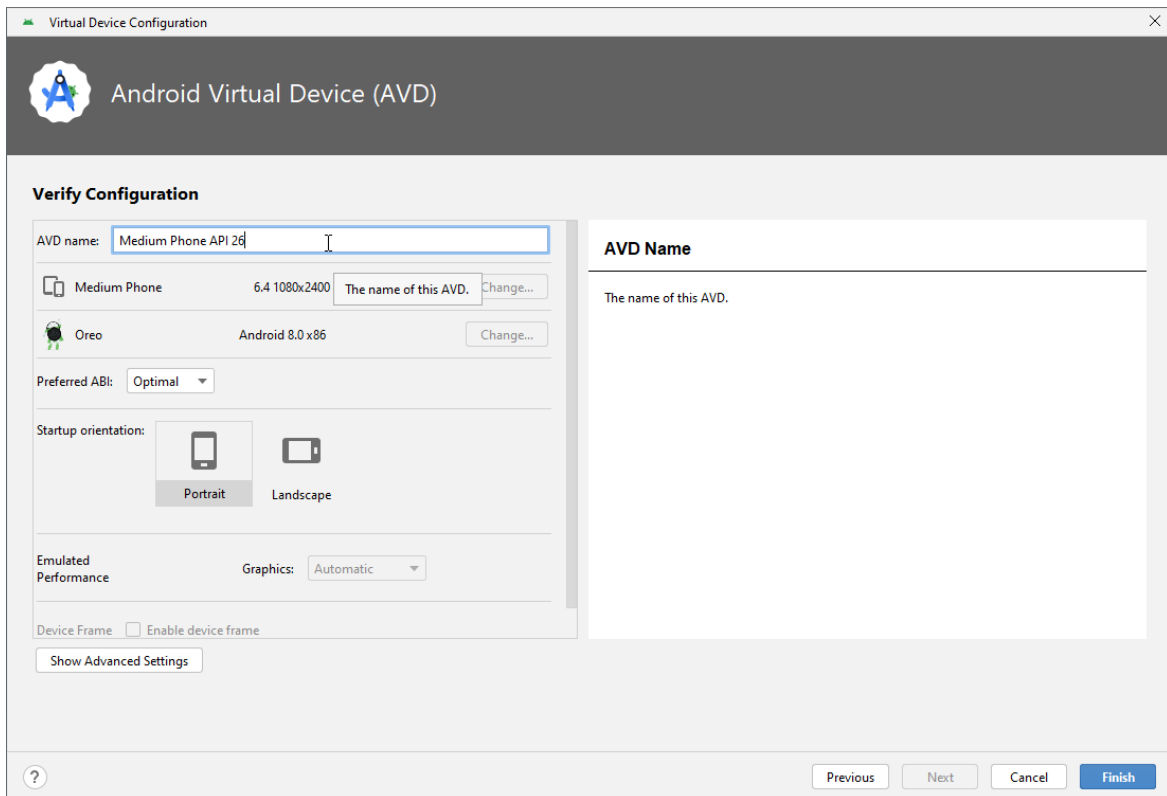


Nu när vi installerat Oreo system image, kan vi använda den i vår emulator.



Introduktion till att skapa appar för android

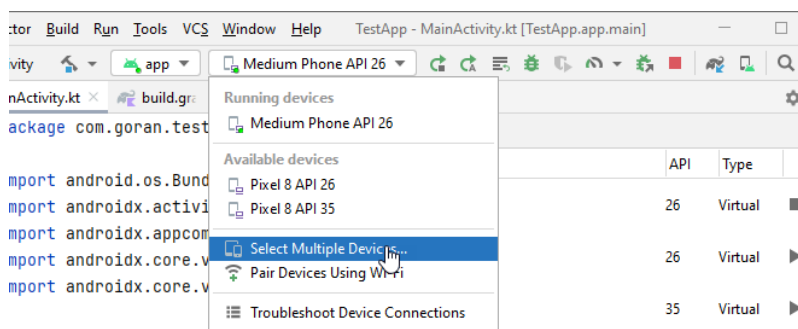
Om du vill, kan du ändra namnet på emulator för att enklare identifiera den. Klicka på *Finish* för att avsluta. Du kan skapa fler emulatorer på samma sätt, om du t ex vill testa dina appar på olika modeller, olika stora skärmar, eller skärmar med olika upplösningar.



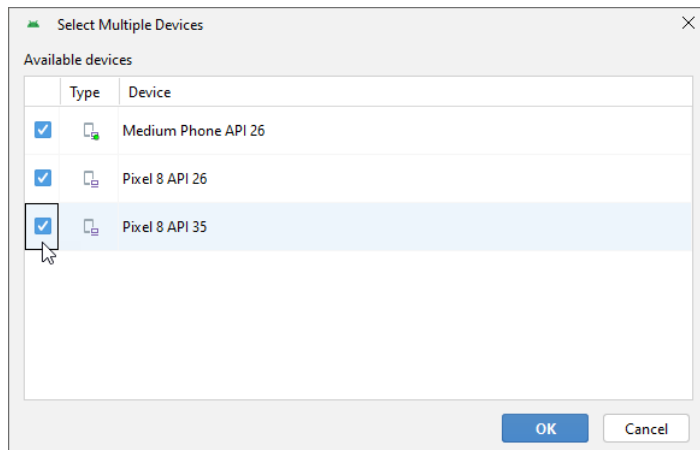
Kör din app på emulatorn

Efter att ha skapat virtuella enheter eller anslutit en fysisk enhet är du redo att starta dina appar inifrån IDE. Du kommer att se att det finns en lista över dina tillgängliga enheter. Du ser både virtuella enheter som du har startat eller fysiska enheter som du har anslutit. Du kommer också att se andra tillgängliga virtuella enheter listade under tillgängliga enheter.

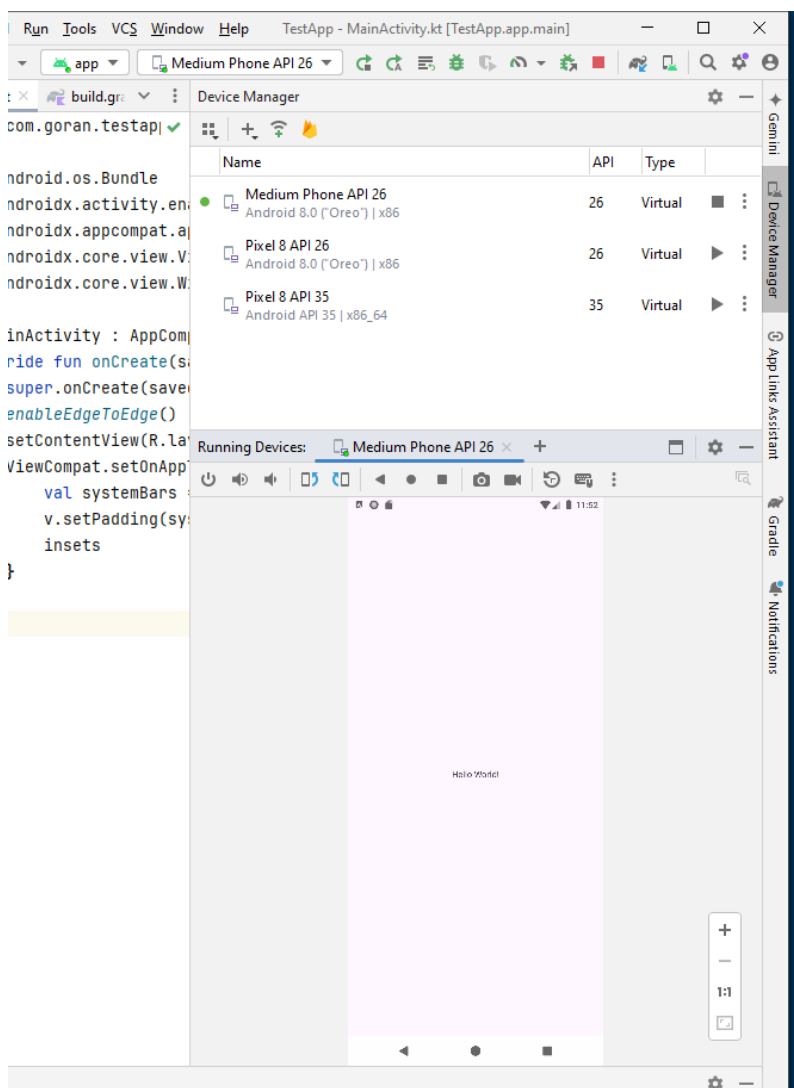
När du kör appen, om du väljer en av dessa, startas den automatiskt. Man kan även välja att köra appen på flera devices (emulatorer och/eller fysiska enheter) samtidigt.



Introduktion till att skapa appar för android

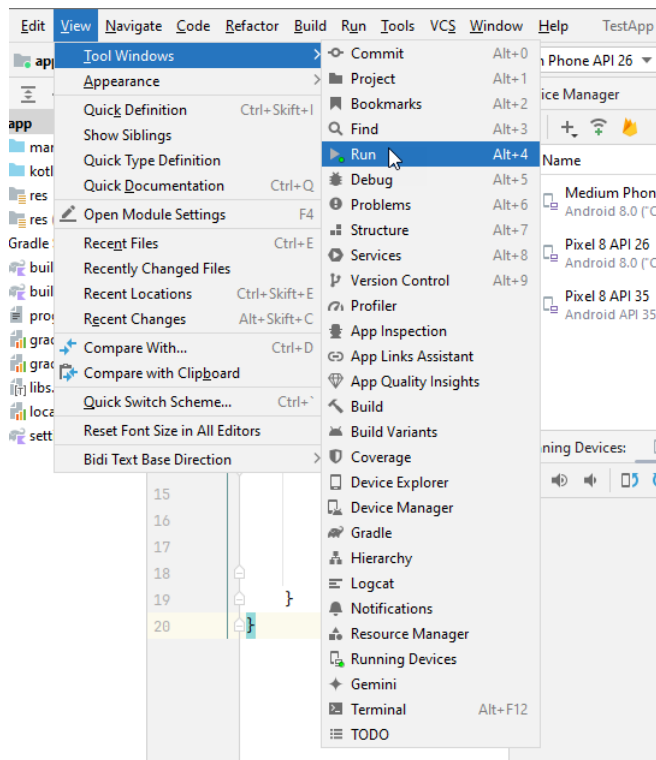


I mitt exempel har jag valt att köra min app på emulatorn *Medium Phone API 26*.



Använd verktygsfönster

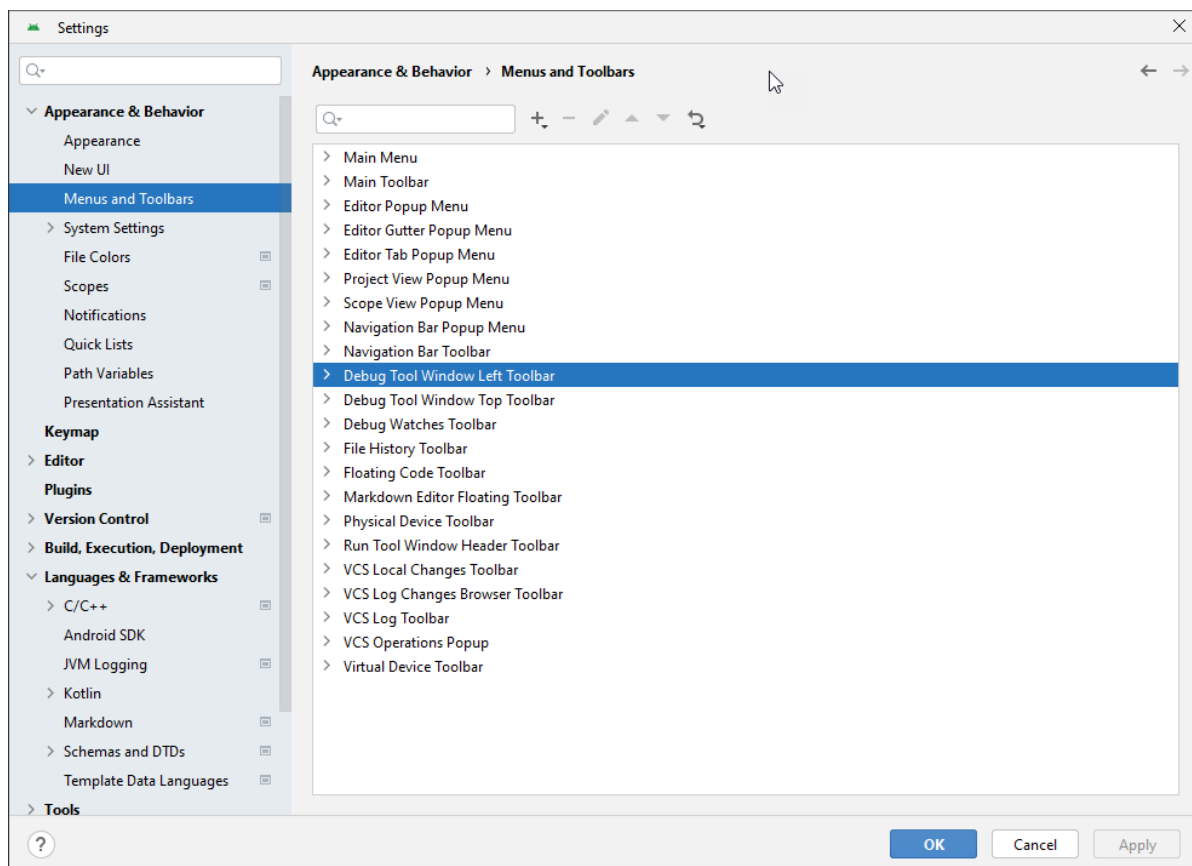
Android Studios användargränssnitt är organiserat kring konceptet fönster. Det primära fönstret är redigeringsfönstret, där du skriver din kod. Men det finns många andra verktygsfönster tillgängliga. Du kan öppna dessa verktygsfönster på flera olika sätt. Från menyn, välj [View > Tool Windows](#) och du kommer att se en omfattande lista över alla fönster som är tillgängliga för dig.



Du kan också komma åt många av dessa fönster genom flikar som omger användargränssnittet. Som standard finns projektfliken uppe till vänster.

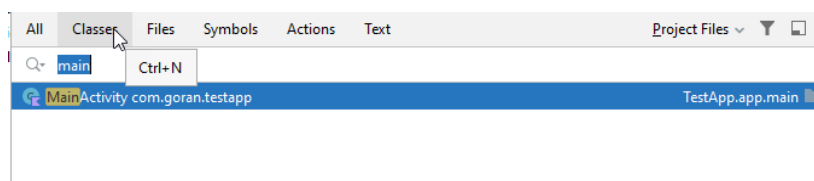
Använd menyer och verktygsfält

Android Studio är baserat på IntelliJ IDEA Community Edition, och det ärver meny- och verktygsfältorganisationen som den produkten har. I Android Studio är standardmenyn baserad på en kombination av IntelliJ IDEA-gränssnittet och anpassade funktioner som är lämpliga för Android. Så till exempel, i IntelliJ IDEA har verktygsmenyn AVD-hanteraren och SDK-hanteraren. Dessa är Android-specifika och de visas inte i IntelliJ IDEA om du inte kör Android-plugin i den produkten. Många av objekten i menyn har associerade tangentbordsgenvägar och du kommer att se dessa tangentbordsgenvägar dokumenterade i menyn själv. Du kan också omorganisera menyn om du vill. Välj [File > Settings > Appearance & Behaviour > Menus and toolbars](#). Här kan du lägga till nya objekt, ta bort objekt eller omorganisera menyn efter dina önskemål.



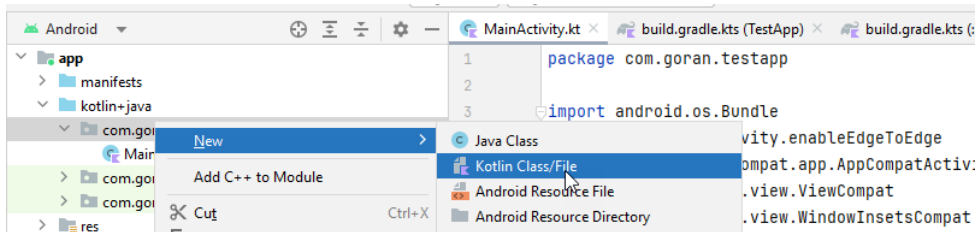
Sök efter filer och kod

Android Studio ger dig många sätt att söka efter dina filer. Det enklaste sättet är att gå till projektfönstret och sedan leta efter filen där. Så till exempel, mina Java- eller Kotlin-filer kommer att ligga under mitt baspaket under Java, men det finns tillfällen när applikationer blir så komplexa att du har mycket djupt nästlade mappar, och det kan ta lite tid att leta efter en fil på det sättet. Så här är ett litet trick. Tryck på shift-tangenten två gånger. End dialogruta dyker upp och du kan börja skriva namnet på den fil i ditt projekt som du söker. Så om jag ville hitta `MainActivity.kt`, skulle jag börja skriva "main" och jag skulle se både `MainActivity.kt` och `activity_main.xml`. Sedan kan jag klicka eller trycka på enter och filen öppnas. Om du vet att det är en klass-fil (`.kt`) då vill du kanske att endast klass-filer som innehåller sträng "main" ska visas – i så fall klickar du på fliken *Classes* högst upp i panelen. Du kan även välja att filtrera endast filer, symboler eller text.

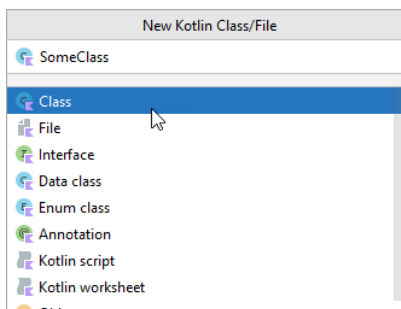


Skapa ny Kotlin fil

För att skapa en ny fil, i projektfönstret, högerklicka på paketet där du vill skapa filen och välj *New > Kotlin Class File*.



Skriv önskat klassnamn i rutan som dyker upp.



OBS! Klassnamn bör

- endast innehålla bokstäver (eventuellt siffror)
- börja med en versal
- börja varje ord med en versal om namnet innehåller flera ord, (t ex `GoransClass`)
- inte innehålla svenska vokaler ä, å, ö

Kotlin är, likt Java, ett objektorienterat språk. För att programmera i Kotlin eller Java på ett effektivt och hållbart sätt kräver att man har goda kunskaper inom objektorienterad programmering. Även om vi i denna kurs kommer att programmera en hel del, så är det inte en ren programmeringskurs; vi lär oss så mycket vi behöver för att kunna skapa androidappar, utan att fördjupa sig i olika programmeringskoncept. Så, sannolikt kommer de flesta av er inte skapa några Kotlin klassfiler, annat än dem som skapas automatiskt när man skapar en `Fragment` eller en `Activity`. Mer om detta kommer senare.