

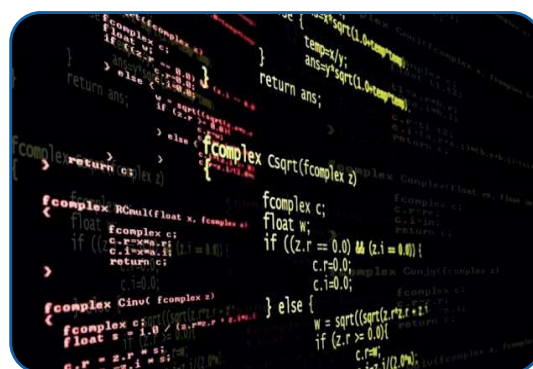
41-dars. MOBIL ILOVALARNI YARATISH TEXNOLOGIYALARI VA VOSITALARI

Har kuni ishlatiluvchi mobil ilovalar qanday yaratilishi haqida hech o'ylab ko'rganmisiz? Aslida, mobil ilova yaratish unchalik ham qiyin ish emas. Demak, mavzu yordamida mobil ilovalar qayerda va qanday yaratilishi mumkinligi bilan tanishamiz.

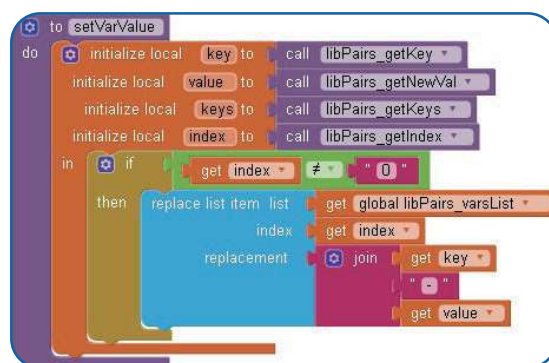
Eng avvalo, mobil ilovalarni yaratish texnologiyalari bilan tanishish zarur. Mobil ilovalar yaratish texnologiyalari, asosan, ikki turga bo'linadi:

- kod orqali dasturlash;
- vizual dasturlash.

Kod orqali dasturlash jarayonida barcha amallar faqat kod yozish orqaligina amalga oshiriladi. Ya'ni dastur yaratish jarayonida dastur oynasini va uning komponentlarini yaratish uchun ham kod yoziladi. Bu usulni ko'proq veb-dasturlashda (HTML, CSS, JavaScript, PHP va b.) va Pythonnig Tkinter kutubxonasida ko'rishimiz mumkin. Ba'zi kod orqali dasturlash platformalarida dastur oynasini yaratish uchun konstruktor rejimlari ham mavjud. Ammo asosiy jarayon kod yozish orqali amalga oshiriladi. Bularga misol tariqasida C# va Delphi dasturlash muhitlarini keltirish mumkin.



Vizual dasturlash – kod yozishning o'rniga grafik obyektlarni boshqarish bilan dastur yaratish usuli. Bunda dasturlash jarayonida maxsus funksiyalardan iborat bloklardan foydalaniladi. Dastur tuzish esa Lego orqali biron buyumni yasash jarayoniga o'xshash bo'ladi. Bu usulda dastur tuzuvchidan dasturlash tillarini yuqori darajada bilish talab qilinmaydi. Ayrim tanikli dastur mualliflari vizual dasturlashni dasturlash tillarini rivojlantirishning keyingi bosqichi (keyingi avlod) deb tan olishadi.



Mobil ilova yaratish imkonini beruvchi ko'plab dasturiy vositalar mavjud. Ularning ayrimlarini ko'rib chiqamiz.

BIG DATA VA DATA MINING

Android Studio



Android Studio – 2013-yil 16-may kuni Google I/O konferensiyasida e’lon qilingan Android platformasi bilan ishlash uchun o’rnatilgan dasturlash muhiti (IDE). IDEni yuklab olish va undan foydalanish mutlaqo bepul. Android studiyada smartfon va planshetlar uchun ilovalar ishlab chiqish vositalari, shuningdek, Android TV, Android Wear, Android Auto, Glass va qo’shimcha kontekstli modullar uchun yangi texnologik yechimlar mavjud.

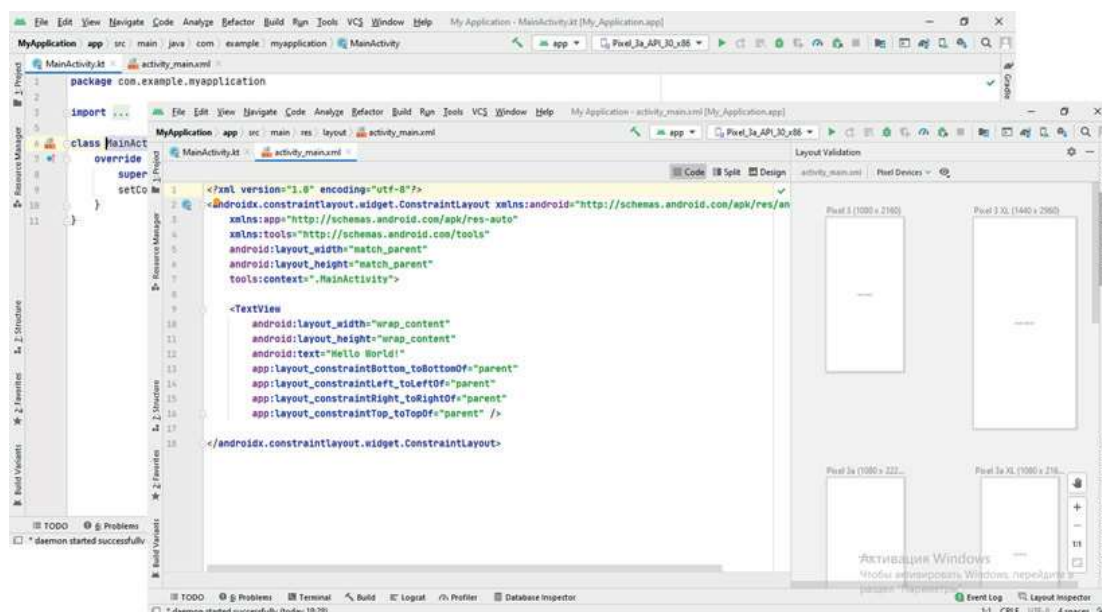
Android Studioda dasturlash 2 muhitda amalga oshiriladi:

- 1) ilova interfeysini yaratish;
- 2) ilova funksiyalarini yaratish.

Ilova interfeysini 2 usulda: kod orqali va komponentlar orqali yaratish mumkin. Kod orqali yaratishda XML tilidan foydalaniladi. Bunda har bir elementning nomi yozilib, so’ngra uning parametrlari sozlanadi. Interfeysni komponentlar yordamida yaratishda esa tayyor komponentlar ilova oynasiga joylanadi va ularning parametrlari sozlanadi. Komponentlarga tugmachalar, yozuv maydonlari, mediamahsulotlarni namoyish qilish maydonlari, veb-maydon kabilar kiradi.

Ilovaning funksiyalarini yaratish dasturlash tillarida amalga oshiriladi. Android Studio muhitida dastlabki versiyalarida dasturlash jarayonlari uchun Java dasturlash tili belgilangan edi.

2017-yil 17-may kuni o’tkazilgan Google I/O konferensiyasida Google Android Studioda Javaga qo’shimcha ravishda Android platformasi uchun rasmiy dasturlash tili sifatida Kotlin tilini qo’llashini e’lon qildi.



Xamarin



Xamarin iOS, Android va Windows qurilmalari uchun C# dasturlash tilida zamonaviy mobil ilovalarni ishlab chiqish freymvorki hisoblanadi.

Xamarin kompaniyasi Amerikada 2011-yilda tashkil etilgan. Uning ishlab chiquvchilari dastlab Microsoft Windows bilan ishlash uchun mo'ljallangan .NET Frameworkni boshqa mobil platformalar uchun moslashtirdilar.

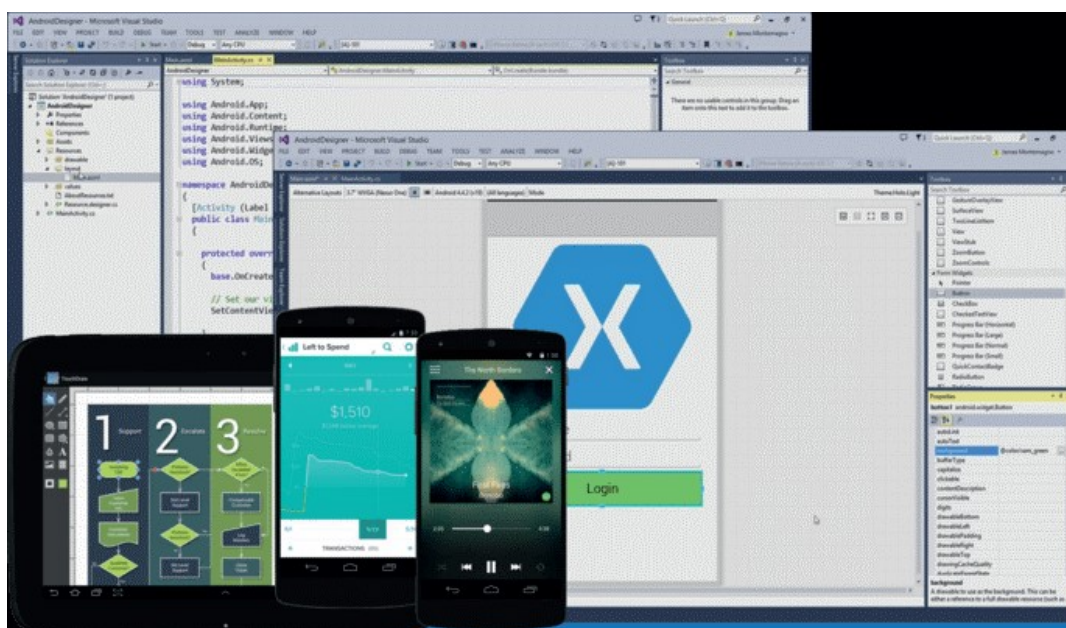
Xamarin .NET bilan zamonaviy, samarali iOS, Android va Windows ilovalarini yaratish uchun ochiq manbali platformadir. Xamarin umumiy kod va asosiy platforma kodlari o'rtasidagi o'zaro aloqalarni boshqaruvchi abstraksiya darajasini taqdim etadi. Xamarin xotira ajratish va kesh fayllarni yig'ish kabi funksiyalarni amalga oshiruvchi boshqariladigan muhitda ishlaydi.

Xamarin bilan o'rtacha 90 % dastur kodi turli platformalarda o'zgarmasdan ishlatilishi mumkin. Ushbu ustuvorlik yordamida dasturchi barcha dastur kodini bitta tilda yozishi (yoki mavjud dastur kodidan foydalanishi) mumkin. Shu bilan birga u mazkur kod orqali turli platformaga xos ishlash imkoniyatiga ega bo'ladi.

Xamarin orqali ilovalarni shaxsiy kompyuterlarda yoki Mac-kompyuterlarida yozilishi va mos platforma paketlariga kompilatsiya qilinishi kerak, masalan, .apk Android uchun yoki .ipa iOS uchun.

Xamarinda ham dasturlash Android Studioda dasturlash kabi ikki muhitda amalga oshiriladi:

- 1) ilova interfeysini yaratish (XAML tili yordamida);
- 2) ilova funksiyalarini yaratish (C# dasturlash tilida).



BIG DATA VA DATA MINING

Thunkable



Thunkable – mobil ilovalar uchun vizual rivojlanish muhiti. “Vizual” soʻzi funktsiya va oʻzgaruvchilarni aks ettiruvchi vizual bloklar bilan dasturlashni anglatadi.

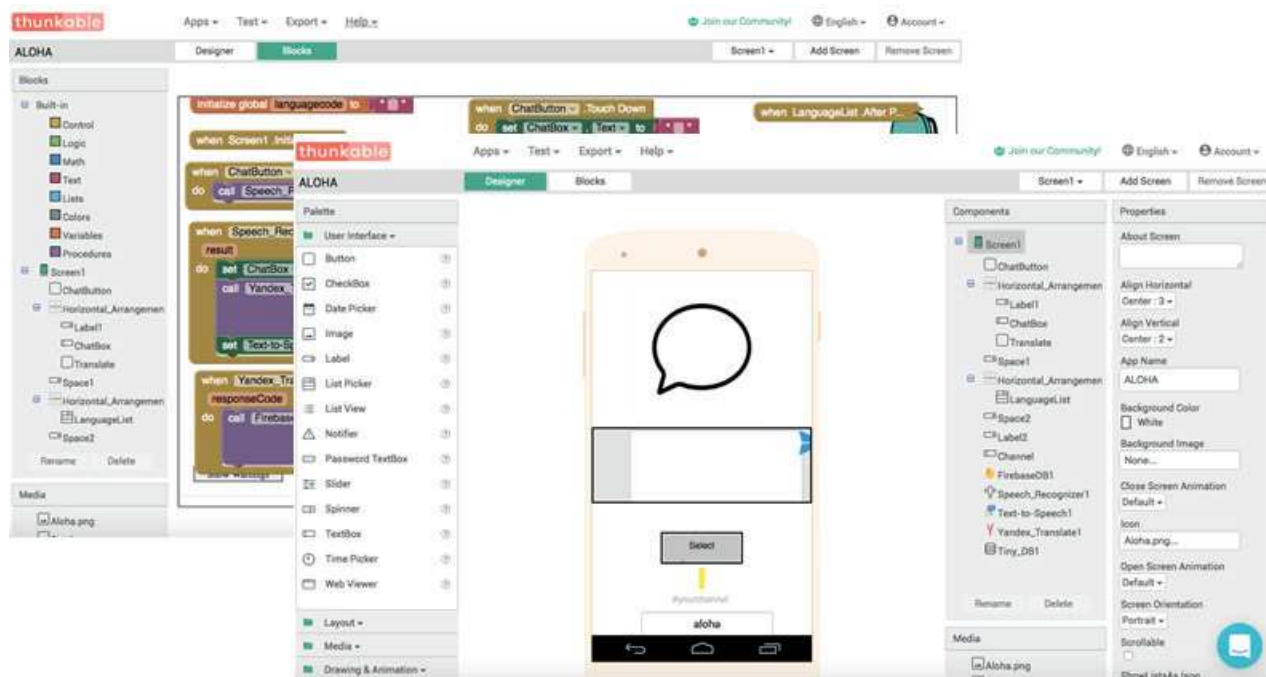
Uning 2 ta asosiy komponenti mavjud:

1) dizayner: ekranga turli xil foydalanuvchi interfeysi komponentlarini qoʻshish orqali dasturning foydalanuvchi interfeysini yaratish imkonini beradi, masalan, tugmachalar, matn maydonlari, GoogleMaps kabilar;

2) bloklar: Thunkableda Scratchga oʻxshash bloklarga asoslangan dasturlash tili mavjud. Bu yangi boshlovchilar uchun dasturlashdan foydalanishni osonlashtiradi, ammo tajribali ishlab chiquvchilar foydalanishi uchun yetarli emas.

Thunkable App Inventor kabi dasturlash muhitlariga oʻxshash tarzda onlayn ishlaydi. Bundan tashqari, bloklardan foydalangan holda vizual dasturlashdan foydalaniladi. Agar istasangiz, masalan, Web Viewer komponenti bilan ishlashda matnli dasturlashdan foydalanishingiz ham mumkin. Ilovani yaratish, nomini oʻzgartirish, nusxalash, oʻchirish va jonli koʻrish bilan bir qatorda, uni App Storeda nashr etish mumkin.

Dasturlash muhitining funksionalligi va Thunkableda dasturlarni yaratish qobiliyati Android uchun kodli dasturlash muhitlaridan sezilarli darajada past, ammo koʻplab vizual va novizual komponentlari yordamida sodda mahalliy va gibriddasturlar yaratish mumkin.



App Inventor



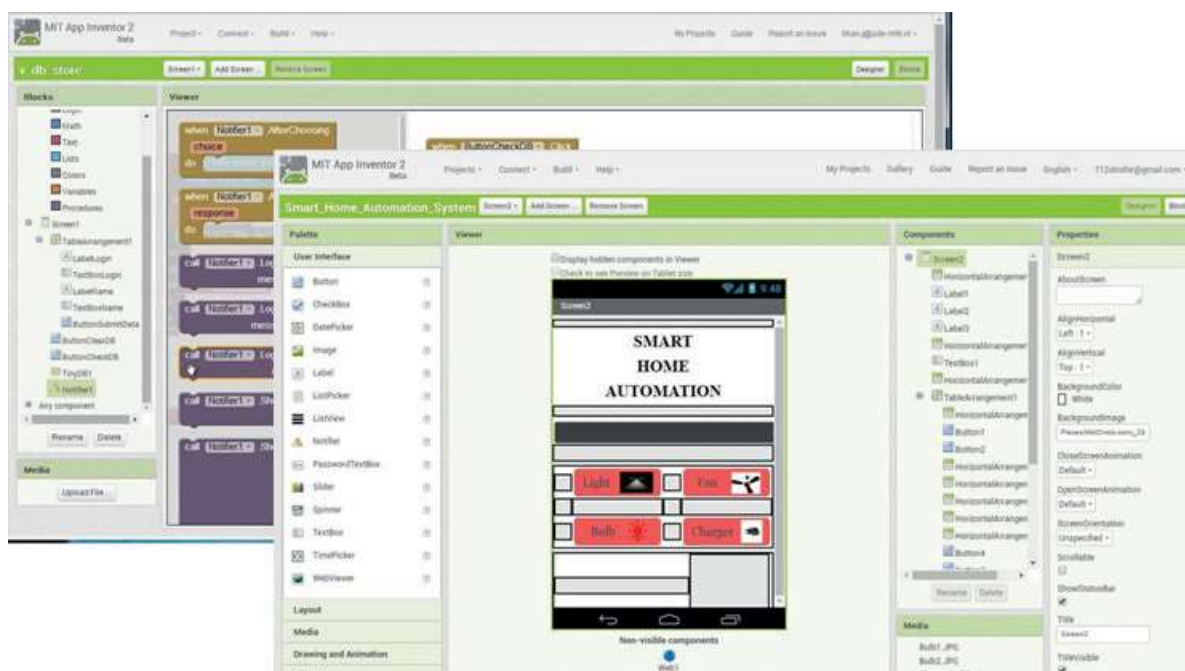
App Inventor (App dastur uchun qisqartirilib, dastur deya tarjima qilingan; Inventor ixtirochi sifatida tarjima qilingan) – dasturlash bo'yicha ko'p bilim talab qilmaydigan vizual dasturlash muhiti. Dastur dastlab Google Labs tomonidan ishlab chiqilgan, keyinchalik esa Massachusetts Texnologiya Institutiga topshirilgan.

App Inventor Android ilovalarni yaratish uchun bulutli texnologiyaga asoslangan vizual dasturlash muhiti hisoblanadi. Dasturlarni qurish dastur kodlari bloklari yordamida vizual rejimda amalga oshiriladi. App Inventor veb-muhitda ishlaydi. Bu esa ishni ancha soddalashtiradi, chunki kompyuter resurslaridan deyarli foydalanilmaydi (lekin muhitga kirish uchun Internet bo'lishi lozim). Shuningdek, Java dasturlash tili va Android SDKni bilish talab qilinmaydi. Kompyuterga hech qanday dasturni o'rnatishi shart emas, faqat Google Chrome brauzerida ai2.appinventor.mit.edu manzili orqali muhitni ishga tushirish mumkin.

App Inventordan foydalanish uchun Google akkaunt zarur. Agar Googleda akkaunt mavjud bo'lmasa, unda ish boshlashdan oldin siz akkaunt yaratishingiz va avtorizatsiya qilishingiz lozim. Shundan so'ng App Inventor dasturi hamda barcha saqlangan loyihalar bilan istalgan kompyuterda ishlash imkoniyati yaratiladi.

Ilovalarni yaratish 2 bosqichda amalga oshiriladi:

- birinchi bosqich – foydalanuvchi interfeysini loyihalash;
- kkinchi bosqich – ilovalarni dasturlash (ko'rsatmalar yaratish).



MOBIL ILOVALAR YARATISH MUHITI



SAVOL VA TOPSHIRIQLAR

1. Mobil ilovalarni yaratishning qanday texnologiyalari mavjud?
2. Kod orqali dasturlashni ta'riflab bering.
3. Vizual dasturlashning afzalligi nimada?
4. Androd Studio dasturi qachon yaratilgan?
5. Xamarin qaysi dasturlash tillariga asoslangan?
6. Thunkable qanday muhitda ishlaydi?
7. App Inventordan foydalanishda eng muhim talab haqida ma'lumot bering.

UYGA VAZIFA

1. Qidiruv tizimi orqali mobil ilovalarni dasturlash muhitlari haqida ma'lumot izlang va ro'yxatini tuzing.
2. Ro'yxatdagi dasturlash muhiti turlari bo'yicha jadvalni to'ldiring:

| Kod orqali dasturlash muhitlari | Vizual dasturlash muhitlari |
|---------------------------------|-----------------------------|
| | |