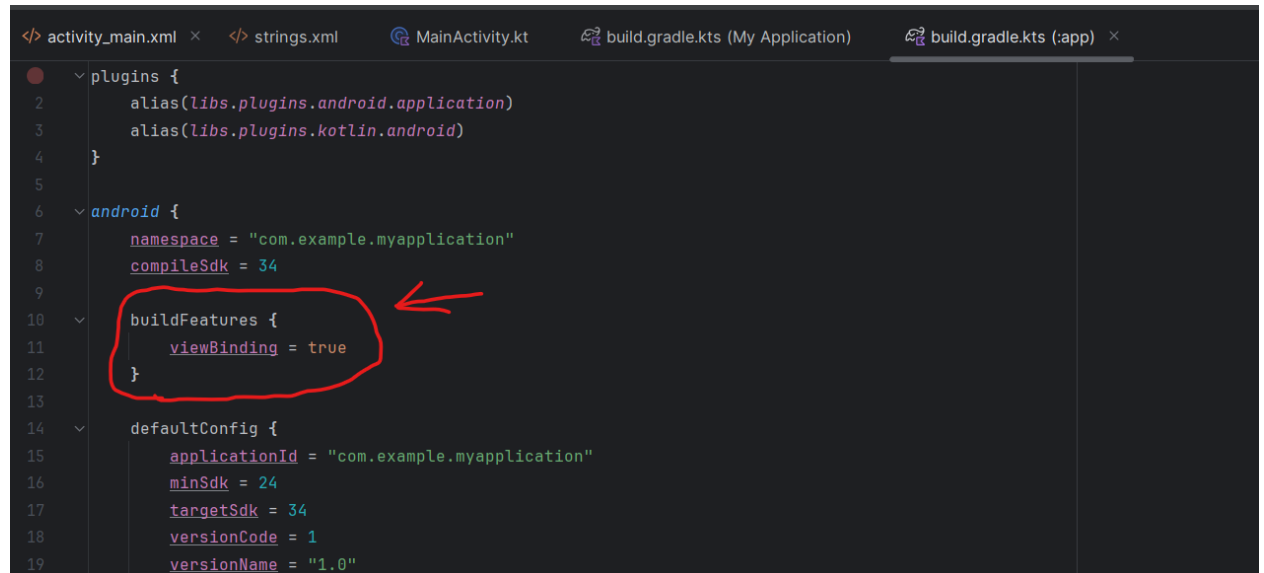


Вопрос: Как работает isChecked()?

используется для проверки состояния элементов интерфейса, которые могут быть включены или выключены, таких как CheckBox, RadioButton и ToggleButton. Этот метод возвращает true, если элемент находится в "включенном" состоянии (отмечен или выбран), и false, если он "выключен" (не отмечен или не выбран).

Скопировал код из пдф к себе и изменил build.gradle.kts. в практике это не было написано, но у меня без этого были ошибки:



```
1 plugins {  
2     alias(libs.plugins.android.application)  
3     alias(libs.plugins.kotlin.android)  
4 }  
5  
6 android {  
7     namespace = "com.example.myapplication"  
8     compileSdk = 34  
9  
10    buildFeatures {  
11        viewBinding = true  
12    }  
13  
14    defaultConfig {  
15        applicationId = "com.example.myapplication"  
16        minSdk = 24  
17        targetSdk = 34  
18        versionCode = 1  
19        versionName = "1.0"  
20    }  
21 }
```

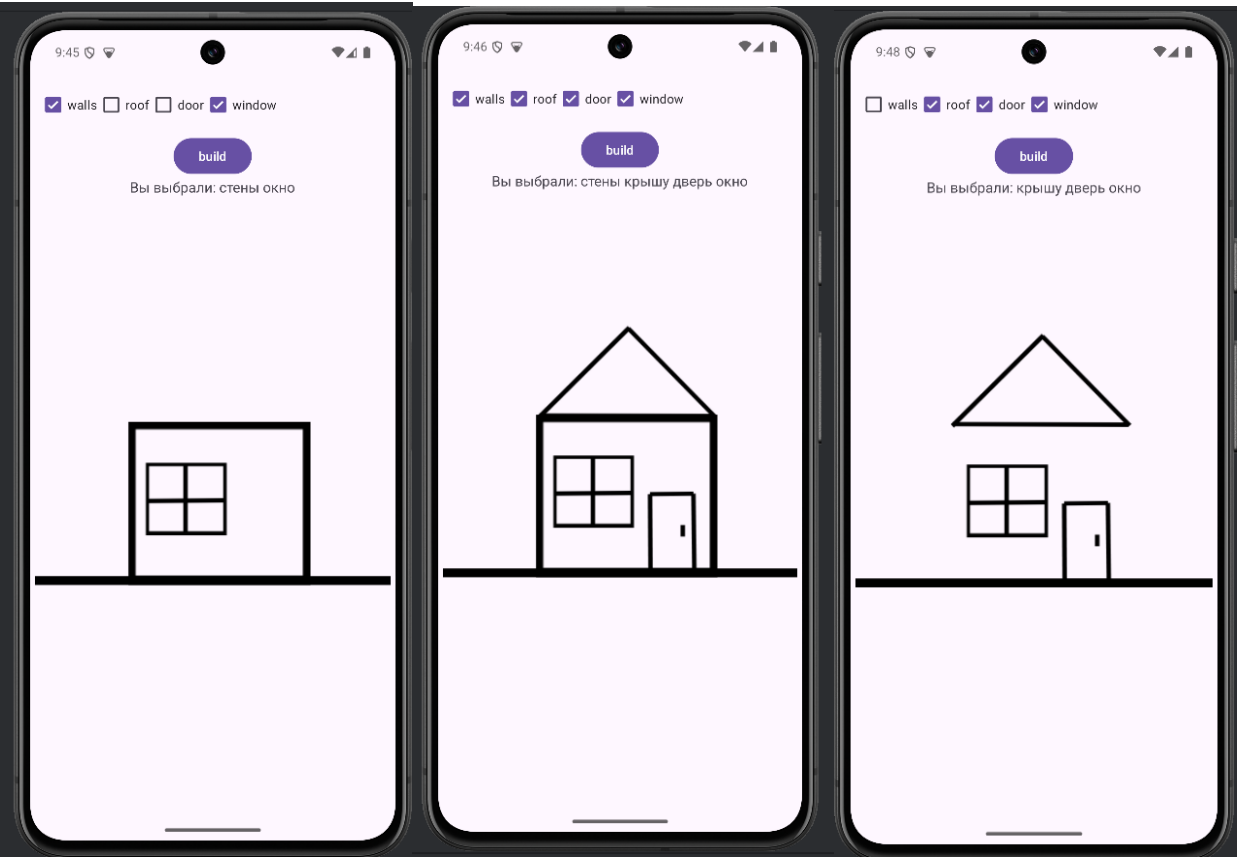
Что получилось:



Моё приложение:
постройка дома



Выбираем то что хотим построить, а после нажимаем build



Код:

xml

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="50dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!-- Горизонтальная панель с CheckBox -->
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="10dp">

        <CheckBox
            android:id="@+id/cbwalls"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/walls"/>

        <CheckBox
            android:id="@+id/cbroof"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/roof"/>

        <CheckBox
            android:id="@+id/cbdoor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/door"/>

        <CheckBox
            android:id="@+id/cbwindow"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/window"/>
    </LinearLayout>

    <Button
        android:id="@+id/btnChoose"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/build" />

    <TextView
        android:id="@+id/tvChoose"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:layout_gravity="center"/>

    <!-- Контейнер для наложения изображений -->
    <FrameLayout
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="140dp">
    >

    <ImageView
        android:id="@+id/imageViewWalls"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:srcCompat="@drawable/walls"
        android:visibility="gone"
    />

    <ImageView
        android:id="@+id/imageViewRoof"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:srcCompat="@drawable/roof"
        android:visibility="gone"
    />

    <ImageView
        android:id="@+id/imageViewDoor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:srcCompat="@drawable/door"
        android:visibility="gone"
    />

    <ImageView
        android:id="@+id/imageViewWindow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:srcCompat="@drawable/window"
        android:visibility="gone"
    />

    <ImageView
        android:id="@+id/imageViewGround"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:srcCompat="@drawable/ground" />
</FrameLayout>

```

```
</LinearLayout>
```

Kt

```

package com.example.myapplication

import android.os.Bundle
import android.view.View
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.example.myapplication.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding

```

```

        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)
            enableEdgeToEdge()
            binding = ActivityMainBinding.inflate(layoutInflater)
            setContentView(binding.root)

            // Обработка системных отступов
            ViewCompat.setOnApplyWindowInsetsListener(binding.main) { v, insets -
>
                val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
                v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
                insets
            }

            // Установка слушателя для кнопки
            binding.btnChoose.setOnClickListener {
                // Скрываем все изображения
                binding.imageViewWalls.visibility = View.GONE
                binding.imageViewRoof.visibility = View.GONE
                binding.imageViewDoor.visibility = View.GONE
                binding.imageViewWindow.visibility = View.GONE

                // Показываем выбранные изображения в зависимости от состояния
CheckBox
                if (binding.cbwalls.isChecked) {
                    binding.imageViewWalls.visibility = View.VISIBLE
                }
                if (binding.cbroof.isChecked) {
                    binding.imageViewRoof.visibility = View.VISIBLE
                }
                if (binding.cbdoor.isChecked) {
                    binding.imageViewDoor.visibility = View.VISIBLE
                }
                if (binding.cbwindow.isChecked) {
                    binding.imageViewWindow.visibility = View.VISIBLE
                }

                // Обновление текста с выбранными элементами
                val result = StringBuilder("Вы выбрали: ")
                if (binding.cbwalls.isChecked) result.append("стены ")
                if (binding.cbroof.isChecked) result.append("крышу ")
                if (binding.cbdoor.isChecked) result.append("дверь ")
                if (binding.cbwindow.isChecked) result.append("окно ")

                binding.tvChoose.text = result.toString().trim()
            }
        }
    }
}

```