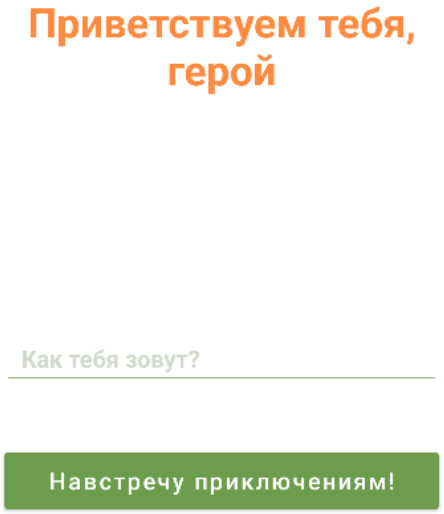
Практическая работа № 6

начиная с этой практической работы, листинги кода могут содержать ошибки или пропуски строчек кода по ранее пройденному материалу

Откройте проект, созданный в 1 практической работе (игровое приложение). Настройте следующую логику для второго экрана - экрана приветствия:



1. Переход на следующий экран должен происходить по нажатию на кнопку
2. Переход будет возможен только если будет заполнено текстовое поле
3. Если текстовое поле не заполнено, должно появляться сообщение об ошибке.

Пользователь вводит имя, это имя должно сохраняться во внутренней памяти приложения. Для этого будет использоваться внутренняя память приложения

Preferences *- настройки - сохранение примитивных данных, значения сохраняются в виде пары: имя (ключ string) и значение. Например: небольшие кусочки данных для дальнейшего использования - данные о пользователе, настройки конфигурации и т.д. значение. В качестве значений могут выступать данные следующих типов: Boolean, Float, Integer, Long, String, набор строк. Настройки могут быть общими для всех activity в приложении, но также могут быть и настройки непосредственно для отдельных activity. Настройки хранятся в xml-файлах в незашифрованном виде в локальном хранилище. Так как настройки хранятся в незашифрованном виде, то не рекомендуется сохранять в них чувствительные данные, например пароль.*

SharedPreferences *— постоянное хранилище на платформе Android, используемое приложениями для хранения своих настроек. Для работы с данными постоянного хранилища необходимо* ***создать экземпляр класса SharedPreferences*** *и задать для него начальное значение. Затем необходимо его* ***инициализировать*** *через метод* ***getSharedPreferences*** *и указать 2 параметра:* ***name*** *— выбранный файл настроек (таблицу, где будут сохраняться под ключевым словом данные, если файл с таким именем не существует, он будет создан при вызове метода edit() и фиксировании изменений с помощью метода commit()) и* ***mode*** *— режим работы -* ***MODE_PRIVATE*** *— доступ к настройкам будет только у этого приложения.*

В одной таблице нельзя использовать один ключ несколько раз. В разных таблицах ключи могут иметь одинаковое имя.

Далее необходимо создать функцию, которая будет записывать данные. В качестве параметра функции необходимо указать тип и количество данных, которые будут записываться.

Откройте файл логики первой активити `MainActivity2.kt` (имя вашей активити может отличаться). Пропишите следующие строки кода (код, представленный на скриншотах ниже, не содержит реализации функционала, прописанного в пунктах с 1 по 3, он относится только к сохранению данных во внутреннюю память приложения):

```
11  class MainActivity2 : AppCompatActivity() {
12      var preff:SharedPreferences?=null
13          lateinit var password:EditText
14          override fun onCreate(savedInstanceState: Bundle?) {
15          super.onCreate(savedInstanceState)
16          setContentView(R.layout.activity_main2)
17      preff=getSharedPreferences( name: "Table", MODE_PRIVATE)
18          }
```

12 строка - для работы с данными постоянного хранилища необходимо создать экземпляр класса SharedPreferences и присвоить ему начальное значение **null**

ключевое слово ***null*** *представляет специальный литерал, который указывает, что* ***переменная (объект) не имеет*** *как такового* ***значения****.*

17 строка - для инициализации переменной preff используется метод **getSharedPreferences**. В скобках указаны 2 параметра: **name — название** файла настроек (название таблицы, где будут сохраняться пары значений ключ - данные) и **mode — режим работы** - MODE_PRIVATE. Все другие режимы работы являются устаревшими и не используются

Затем необходимо прописать функцию сохранения данных в таблицу. Функции будет передаваться один параметр - имя переменной, значение которой необходимо

сохранить.

```
11  class MainActivity2 : AppCompatActivity() {
12      var preff:SharedPreferences?=null
13          lateinit var password:EditText
14          override fun onCreate(savedInstanceState: Bundle?) {
15          super.onCreate(savedInstanceState)
16          setContentView(R.layout.activity_main2)
17      preff=getSharedPreferences( name: "Table", MODE_PRIVATE)
18          }
19      fun savedata(name:String)
20      {
21      val editor=preff?.edit()
22          editor?.putString("key1", name)
23          editor?.apply()
24
25      }
```

19 строка - объявление функции

20 - 25 строки - тело функции

21 строка - доступ к редактированию таблицы данных через объявление дополнительной переменной

22 строка - записываем значение переменной name в таблицу, используя для записи ключ key1 (имя ключа определяете самостоятельно, это всегда значение типа String - текст - указывается в кавычках), используя метод putString (в зависимости от типа записываемого значения, указанного в параметрах функции, метод будет меняться) :

- putBoolean(String key, boolean value),
- putFloat(String key, float value),
- putInt(String key, int value),
- putLong(String key, long value),
- putString(String key, String value),
- putStringSet(String key, Set values)

По имени ключа в дальнейшем можно будет получить доступ к записанным данным.

Сохранение имени будет происходить при нажатии на кнопку, как только пользователь нажмет на кнопку Навстречу приключениям!, функция savedata должна исполняться.

```
18      preff=getSharedPreferences( name: "Table", MODE_PRIVATE)
19          }
20      fun savedata(name:String)
21      {
22      val editor=preff?.edit()
23          editor?.putString("key1", name)
24          editor?.apply()
25      }
26
27      fun nextt(view: View) {
28          val username:String=name.text.toString()
29          savedata(username)
30      }
31  }
```

27 строка - функция обработки нажатия на кнопку. Обратите внимание, на изображении отсутствует код проверки на пустоту текстового поля и перехода на следующий экран. Данный функционал вы прописываете самостоятельно.

28 строка - объявляем переменной и присвоение ей значения параметра text текстового поля

29 строка - вызов функции сохранения имени пользователя в таблицу внутренней памяти приложения, в качестве аргумента функции передана переменная username, в которую было записано, какой текст пользователь ввел в текстовое поле. Самостоятельно допишите недостающие строки кода.

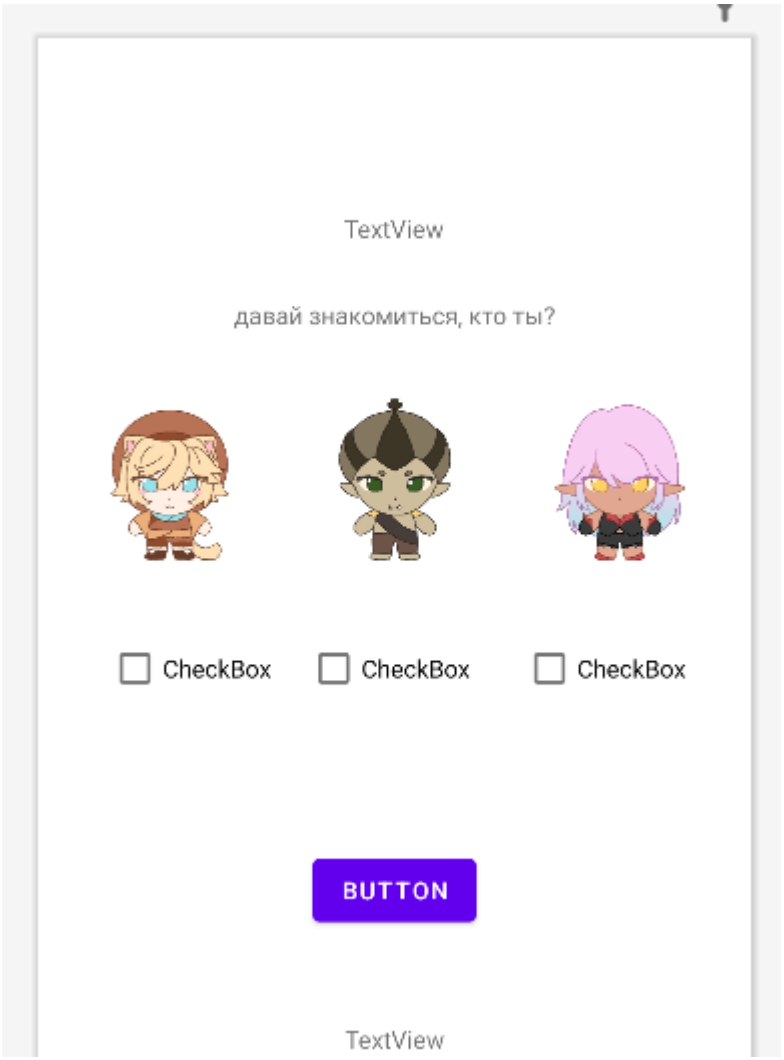
Запустите приложение, проверьте как оно работает.

На следующем экране должно отображаться приветствие:

имя_пользователя

давай знакомиться, кто ты?

Измените макет экрана



В контейнере TextView должно отображаться имя пользователя. Для этого необходимо считать имя пользователя из таблицы:

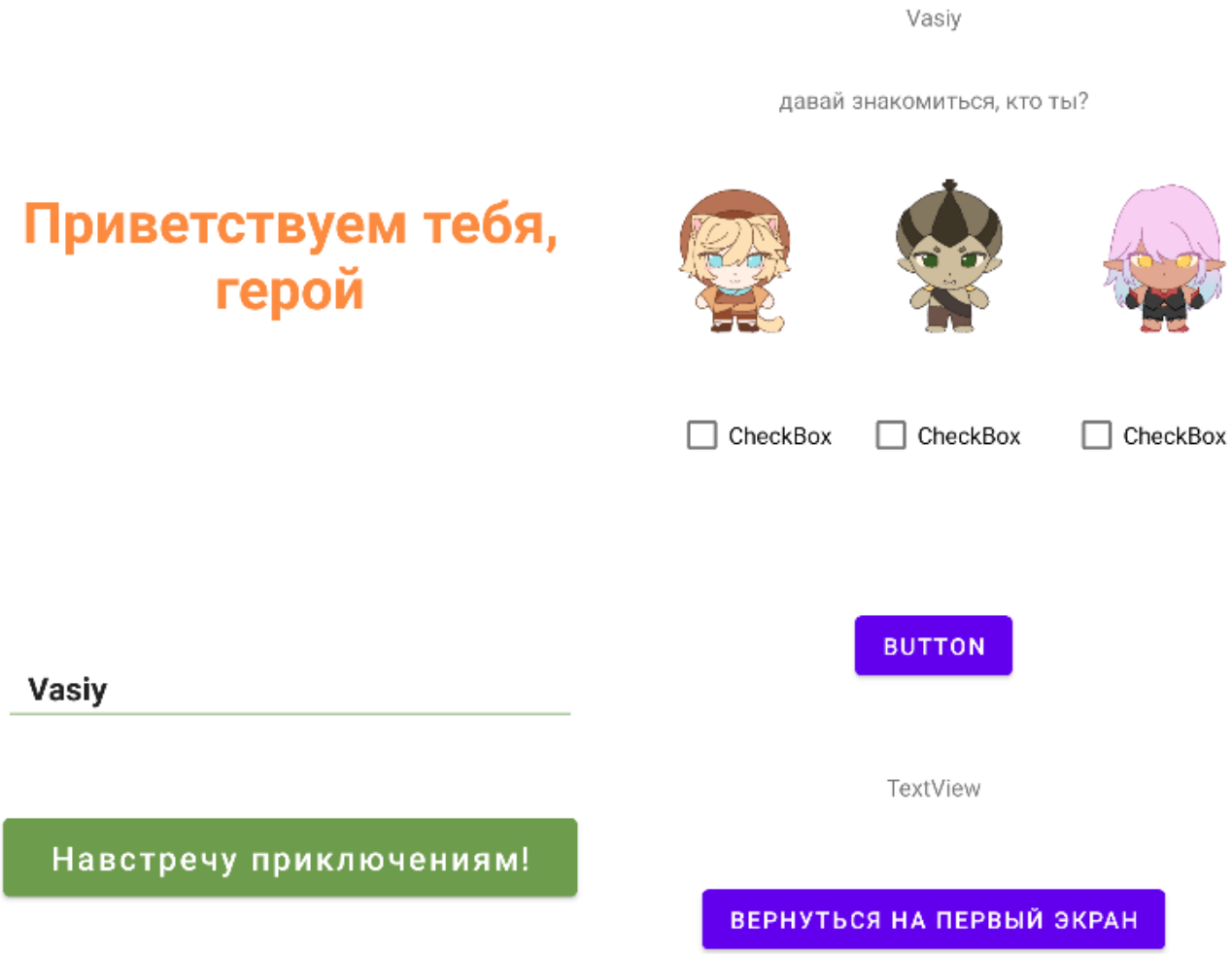
```
10  class MainActivity3 : AppCompatActivity() {
11      lateinit var prefff:SharedPreferences
12      lateinit var name:TextView
13      override fun onCreate(savedInstanceState: Bundle?) {
14          super.onCreate(savedInstanceState)
15          setContentView(R.layout.activity_main3)
16          name=findViewById(R.id.textView2)
17          prefff=getSharedPreferences( name: "Table", MODE_PRIVATE)
18  name.text=prefff?.getString( key: "key1", defValue: "")
19      }
```

11 строка - создать экземпляр класса SharedPreferences и не присвоить ему начальное значение **null,** мы будем его инициализировать с уже существующей таблицей

17 строка - инициализация prefff с таблицей Table, созданной и заполненной на предыдущем экране

18 строка - вывод текста в контейнере TextView (name.text), текст для вывода будет получен (getString) из таблицы Table по ключу key1, если ключ key1 не имеет значения, ничего не будет выведено на экран, контейнер останется пустым.

Еще раз запустите приложение:

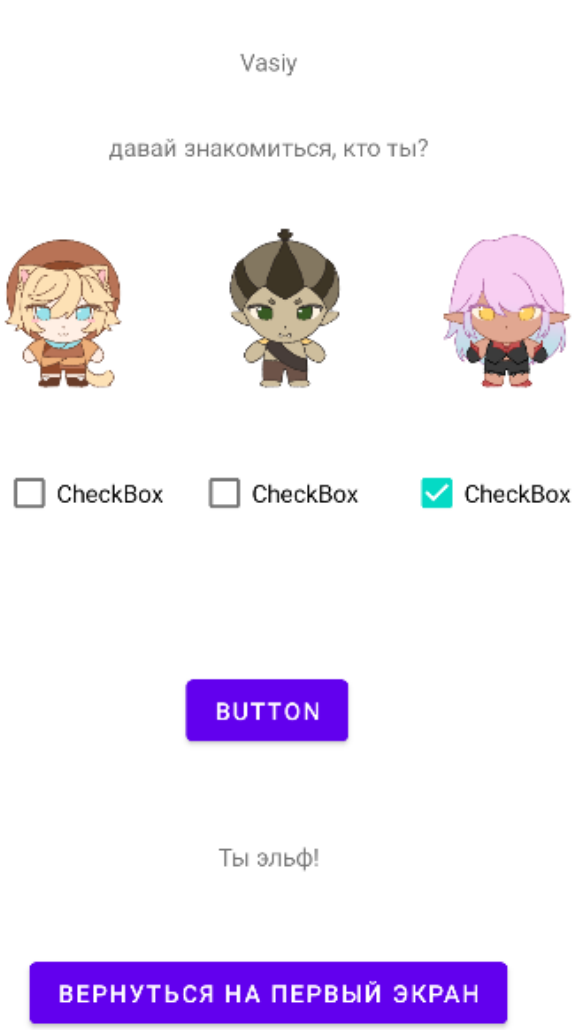


При нажатии на кнопку Button, если пользователь отметит первый чек-бокс, должна появиться надпись Ты нэко!, если второй - Ты гном!, если третий - Ты эльф!.

```
class MainActivity3 : AppCompatActivity() {
    lateinit var prefff:SharedPreferences
    lateinit var name:TextView
    lateinit var cb1:CheckBox
    lateinit var cb2:CheckBox
    lateinit var cb3:CheckBox
    lateinit var race:TextView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main3)
        name=findViewById(R.id.textView2)
        cb1=findViewById(R.id.checkBox)
        cb2=findViewById(R.id.checkBox2)
        cb3=findViewById(R.id.checkBox3)
        race=findViewById(R.id.textView3)
        prefff=getSharedPreferences( name: "Table", MODE_PRIVATE)
name.text=prefff?.getString( key: "key1", defValue: "")
    }

    fun proverka(view: View) {
        if(cb1.isChecked==true)
        {
            race.text="Ты нэко!"
        }
        else {
            if (cb2.isChecked==true)
            {
                race.text="Ты гном!"
            }
```

31 строка - параметр .isChecked проверяет нажат или нет чек-бокс и возвращает истину, если элемент был отмечен и ложь, если нет. Напишите код для проверки нажатия на все три чек-бокса.



Настройте оформление данного экрана.