

1. Реализовать Java-программу в NetBeans IDE демонстрирующую принципы ООП (наследование, инкапсуляция, полиморфизм) на примере иерархии классов сущности Animal (в иерархии должно быть 3 уровня).

```
public class OOP_Princip_1 {
    // В ООП Инкапсуляция реализуется через наследование, виртуальные методы и абстрактные классы. Данные, связанные с объектом,
    // должны быть скрыты от внешнего мира. Позволяет зашитить данные от изменения.
    // Полиморфизм позволяет объектам разных типов иметь одинаковые методы.
    public static void main(String[] args) {
        System.out.println("Вопрос 1. Демонстрация принципов ООП на примере иерархии классов сущности Animal.");
        Animal[] animals = {
            new Dog(name: "Akita", age: 5), new Cat(name: "Main Coon", age: 4)
        };

        for (Animal anim : animals) { //проходим по массиву animals.
            System.out.printf(format: "%s (%d years old) says '%s'\n", args: anim.getName(), args: anim.getAge(), args: anim.makeSound());
            //System.out.println(); метод для форматированного вывода данных на консоль.
            // %s - String, %d - целое число, \n - перевод строки на следующую.
            System.out.println();
        }
    }
}

package com.mycompany.oop_princip_1;

public abstract class Animal { //Абстрактный метод определяет ОБЩУЮ структуру для всех животных. включая методы makeSound(),
    //которые не имеют конкретного значения для каждого ОТДЕЛЬНОГО вида животного.
    //Абстрактные классы содержат только методы без реализации. которые должны быть переопределены
    //в дочерних классах для создания конкретных объектов.
    //Animal это 1 уровень иерархии.
    private String name;
    private int age;

    public Animal(String name, int age){
        this.name = name;
        this.age = age;
    }

    public String getName(){
        return name;
    }

    public int getAge(){
        return age;
    }

    public abstract String makeSound(); //это тот самый полиморфизм
}

public abstract class Mammal extends Animal{ //Mammal это 2 уровень иерархии.
    public Mammal(String name, int age){
        super(name, age); //здесь super - обращение к переменным name и age родительского класса
    }

    public abstract void pet(); //класс Mammal также является абстрактным и добавляет метод pet() для домашних животных.
    //Если в абстрактном классе определен абстрактный метод, то ВСЕ наследники должны этот абстрактный метод переопределить.
}

public class Cat extends Mammal{ //3 уровень иерархии.
    public Cat(String name, int age){
        super(name, age);
    }

    @Override
    public String makeSound(){ //String показывает, какое значение мы хотим вернуть. "Meow" - это String переменная.
        return "Meow";
    }

    @Override
    public void pet(){
        System.out.println(getName() + "is a pet.");
    }
}

public class Dog extends Mammal { //3 уровень иерархии.
    public Dog(String name, int age){
        super(name, age);
    }

    @Override
    public String makeSound(){ //String показывает, какое значение мы хотим вернуть.
        return "Woof";
    }

    @Override
    public void pet(){
        System.out.println(getName() + "is a pet.");
    }
}
```

2. Реализовать Java-программу в NetBeans, которая преобразовывает введенную пользователем с клавиатуры строку.

```

public class String_StringBuilder_2 {

    public static void main(String[] args) {
        System.out.println("Задание 2. Java-программа, которая преобразовывает введенную пользователем строку");
        Scanner scanner = new Scanner(System.in);

        System.out.println("Введите строку на английском: ");
        String str = scanner.nextLine();

        System.out.println("Оригинал: " + str);
        System.out.println("Развернутая строка: " + StringUtils.reverse(str));
        System.out.println("Нижний регистр: " + StringUtils.toLowerCase(str));
        System.out.println("Верхний регистр: " + StringUtils.toUpperCase(str));
        System.out.println("Удаление гласных: " + StringUtils.removeVowels(str));
    }

    public class StringUtils {

        public static String reverse(String str){ //Разворот строки.
            StringBuilder sb = new StringBuilder();
            for (int i = str.length() - 1; i >= 0; i--){
                sb.append(str.charAt(i));
            }
            return sb.toString();
        }

        public static String toLowerCase(String str){ //Нижний регистр.
            return str.toLowerCase();
        }

        public static String toUpperCase(String str){ //Верхний регистр.
            return str.toUpperCase();
        }

        public static String removeVowels(String str){ //Удаление гласных.
            return str.replaceAll("[aeiouAEIOU]", "");
        }
    }
}

```

```

Задание 2. Java-программа, которая преобразовывает введенную пользователем строку
Введите строку на английском:
Hello, my friends
Оригинал:Hello, my friends
Развернутая строка: sdneirf ym ,olleH
Нижний регистр: hello, my friends
Верхний регистр: HELLO, MY FRIENDS
Удаление гласных: Hll, my frnds

```

Найти все методы работы со String.

3. Реализовать Java-программу в NetBeans, которое вычисляет максимальное и минимальное число из введенных пользователем с клавиатуры числовых значений через запятую. в решении необходимо использовать цикл `for`. Проверки на некорректные значения – обязательны!

```

public static void main(String[] args) {
    System.out.println("Java-программа в NetBeans, которая вычисляет max и min число из введенных пользователем через запятую значений. ");
    Scanner scanner = new Scanner(System.in);
    System.out.println("Введите числа через запятую: ");
    String strNumbers = scanner.nextLine(); //считываем строку.
    String[] numbers = strNumbers.split(","); //очищаем строку от запятых.
    List<Integer> list = new ArrayList(); //создаём лист в которой будем добавлять числа.

    Integer[] num = new Integer[numbers.length]; // создаём числовой массив и выделяем место равное размеру длины введенной строки.
    for(int i = 0; i < numbers.length; i++){
        try {
            num[i] = Integer.valueOf(numbers[i]); //преобразуем String значения в Integer.
        } catch (NumberFormatException nfe) //перехватываем ошибку в случае, если будет введено не число.
        {continue;}
        list.add(num[i]);
    }

    Collections.sort(list); //Сортируем массив по возрастанию.
    int c = list.size()-1;
    System.out.println("max число " + list.get(c));
    System.out.println("min число " + list.get(0));
}

```

Java-программа в NetBeans, которая вычисляет max и min число из введенных пользователем через запятую значений.
Введите числа через запятую:
456,456,4,5,9,65,3,ghrty,67,456
max число 456
min число 3

4. Реализовать Java-программу в Apache NetBeans IDE, выполняющую вывод в консоль отсортированных по возрасту сущностей типа Person, (объекты хранятся в ArrayList). В решении необходимо использовать интерфейс Comparable. Заполнение коллекции происходит в коде программы (минимум 3 элемента коллекции).

```

import java.util.ArrayList;
import java.util.Collections;

public class Sort_to_age_4 {

    public static void main(String[] args) {
        System.out.println("Java-программа, выполняющая вывод в консоль отсортированных по возрасту сущностей типа Person из ArrayList.");

        ArrayList<Person> person = new ArrayList<>(); // создаём массив из людей, с данными из класса Person (name, age).
        person.add(new Person( name:"John", age:23));
        person.add(new Person( name:"Mad", age:43));
        person.add(new Person( name:"Sara", age:5));

        Collections.sort( list:person); // сортировка массива person.
        for (Person pers : person){
            System.out.printf( format:"%s (%d)\n", args:pers.getName(), args:pers.getAge());
        }
    }
}

public class Person implements Comparable<Person> { //Person реализует интерфейс Comparable, который позволяет сравнивать объекты.
    private String name;
    private int age;

    public Person(String name, int age){
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    @Override
    public int compareTo(Person other){ // метод compareTo() возвращает результат сравнения возрастов двух объектов типа Person.
        return Integer.compare( this.age, other.age);
    }
}

```

5. Реализовать Java-программу в Apache NetBeans IDE, выполняющую вывод в консоль отсортированного по значению ключа HashMap (ключ – типа String). Инициализировать HashMap допускается напрямую в коде программы.

```

public class Sort_hashmap {

    public static void main(String[] args) {
        System.out.println("Java-приложение для сортировки HashMap по значению ключа в формате String.");

        HashMap<String, Integer> map = new HashMap<>(); //создаём HashMap.
        map.put( key:"apple", value:4);
        map.put( key:"orange", value:5);
        map.put( key:"peach", value:5);

        Map<String, Integer> sortedMap = new TreeMap<>(map); //TreeMap сортирует HashMap по ключу.

        System.out.println(sortedMap);
    }
}

Java-приложение для сортировки HashMap по значению ключа в формате String.
{apple=4, orange=5, peach=5}

```

6. Реализовать Java-программу в Apache NetBeans IDE, сохраняющую в файл строку, которую введет пользователь с клавиатуры. Путь к файлу допускается задать в коде программы.

```

public class String_to_file_6 {

    public static void main(String[] args) {
        System.out.println("Java-приложение для сохранённой введённой пользователем с клавиатуры строки в файл.");

        Scanner scanner = new Scanner(System.in);

        System.out.println("Введите строку, которую нужно сохранить в файл: ");
        String input = scanner.nextLine();

        try (FileWriter wr = new FileWriter("output.txt")){ //FileWrite класс, позволяющий записывать данные файл.
            wr.write(input); //записываем данные в файл. Файл сохраняется в папку проекта. НАЗВАНИЕ ПРОЕКТА/output.txt.
            System.out.println("Строка сохранена в файл.");
        } catch (IOException e) {
            System.out.println("Ошибка сохранения файла: " + e.getMessage());
        }
    }
}

```

7. Реализовать Java-программу в Apache NetBeans IDE, которая загружает содержимое текстового файла, делает над ним некоторые преобразования (например, разворот строки) и выводит полученное значение в консоль. Путь к файлу вводит пользователь с клавиатуры. Проверки на некорректные значения – обязательны!

```

public static void main(String[] args) {
    System.out.println("Java-приложение для считывания строки из файла и проделывания с ней махинаций.");

    Scanner scanner = new Scanner(System.in);
    System.out.println("Введите путь к файлу: ");
    String str = scanner.nextLine();
    File file = new File(str);
    if (!file.exists()) {
        System.out.println("Файл не найден.");
        return;
    }

    try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
        String line = reader.readLine();
        while (line != null) {
            String reversed = (String) reverse(str: line);
            System.out.println(reversed);
            line = reader.readLine();
        }
    } catch (IOException e) {
        System.out.println("Ошибка чтения файла.");
        e.printStackTrace();
    }
}

private static String reverse(String str) {
    return new StringBuilder(str).reverse().toString();
}

```

Путь указывается в формате:

D:\Netbeans\NetBeans-15\Projects\string_to_file_6\output.txt.

8. Реализовать Java-программу в Apache NetBeans IDE, которая сериализует в файл экземпляр класса сущности Person или Animal. Путь к файлу вводит пользователь с клавиатуры.

```

public static void main(String[] args) {
    System.out.println("Java-приложение, которое сериализует в файл экземпляр класса сущности Person или Animal.");
    Scanner scanner = new Scanner(System.in);
    System.out.print("Введите путь к файлу: ");
    String str = scanner.nextLine();

    Person person = new Person(name: "John", age: 30);
    // Animal animal = new Animal("Cat", "Whisky"); //в зависимости от задания может быть Animal.

    try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(str))) {
        out.writeObject(person); //записывает на диск экземпляра класса.
        // out.writeObject(animal);
        System.out.println("Объект " + person + " успешно записан в файл " + str);
    } catch (IOException e) {
        System.out.println("Ошибка записи в файл: " + str);
        e.printStackTrace();
    }
}

```

```

class Person implements Serializable {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    @Override
    public String toString() {
        return "Person{" +
            "name='" + name + '\'' +
            ", age=" + age +
            "'";
    }
}

```

```

class Animal implements Serializable {
    private String type;
    private String name;

    public Animal(String type, String name) {
        this.type = type;
        this.name = name;
    }

    @Override
    public String toString() {
        return "Animal{" +
            "type='" + type + '\'' +
            ", name='" + name + '\'' +
            "'";
    }
}

```

9. Реализовать Java-программу в Apache NetBeans IDE, которая вычисляет площадь прямоугольника в отдельном потоке и выводит результат в консоль. Создание потока должно быть реализовано с использованием интерфейса Runnable. Значения сторон вводит пользователь с клавиатуры. Проверки на некорректные значения – обязательны!

```

public static void main(String[] args) {
    System.out.println("Java-программа, которая вычисляет площадь прямоугольника в отдельном потоке и выводит результат в консоль.");

    Scanner scanner = new Scanner(System.in);

    double a, b;

    while (true) {
        System.out.print("Введите сторону a: ");
        if (scanner.hasNextDouble()) {
            a = scanner.nextDouble();
            break;
        } else {
            System.out.println("Некорректное значение стороны a!");
            scanner.next();
        }
    }

    while (true) {
        System.out.print("Введите сторону b: ");
        if (scanner.hasNextDouble()) {
            b = scanner.nextDouble();
            break;
        } else {
            System.out.println("Некорректное значение стороны b!");
            scanner.next();
        }
    }

    Runnable task = new RectangleAreaCalculator(a, b);
    Thread thread = new Thread(task); //создаём поток.
    thread.start(); //запускаем поток, выполняющий метод run().
}

public class RectangleAreaCalculator implements Runnable { //Runnable интерфейс для запуска потока.
    private double a, b;

    public RectangleAreaCalculator(double a, double b) {
        this.a = a;
        this.b = b;
    }

    public void run() {
        double area = a * b;
        System.out.println("Площадь прямоугольника: " + area);
    }
}

```

10. Реализовать Java-программу в Apache NetBeans IDE, которая вычисляет в отдельном потоке факториал введенного пользователем числа с клавиатуры и выводит полученное значение в консоль. Необходимо использовать цикл for. Создание потока должно быть реализовано с использованием наследника класса Thread. Проверки на некорректные значения – обязательны!

```

public static void main(String[] args) {
    System.out.println("Java-программа, которая вычисляет \n" +
        "в отдельном потоке факториал введенного пользователем числа с клавиатуры \n" +
        "и выводит полученное значение в консоль.");
    Scanner scanner = new Scanner(System.in);

    int num;

    while (true) { // while будет выполняться до тех пор. пока не произойдет какое-то условие.
        //Здесь он прерывается в случае, если введено не число.
        System.out.print("Введите число: ");
        if (scanner.hasNextInt()) {
            num = scanner.nextInt();
            break;
        } else {
            System.out.println("Некорректное значение числа!");
            scanner.next();
        }
    }

    Thread thread = new FactorialCalculator(num); //создание экземпляра Thread на базе объекта FactorialCalculator.
    thread.start(); // запуск нового потока.
}

public class FactorialCalculator extends Thread {
    private int num;

    public FactorialCalculator(int num) {
        this.num = num;
    }

    public void run() {
        int factorial = 1;
        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }
        System.out.println("Факториал " + num + " = " + factorial);
    }
}

```

12. Реализовать мобильное приложение в Android Studio, которое складывает два целых числа, введенных пользователем в поля ввода. Результат вычисления необходимо отобразить в виджете TextView после нажатия на кнопку (Button). В макете activity разрешено использовать контейнеры только одного типа: LinearLayout. Проверки на некорректные значения – обязательны!

Создаём <LinearLayout> ВНУТРИ ДРУГОГО <LinearLayout>! <Button/> можно внутри любого контейнера из двух.

```
<LinearLayout
    android:layout_marginTop="40dp"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/tvResult"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/tvValue1"
        android:layout_weight="1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:inputType="number"
        android:textSize="12dp"
        android:hint="Введите 1 слагаемое"
    />

    <EditText
        android:id="@+id/tvValue2"
        android:textSize="12dp"
        android:layout_weight="1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:inputType="number"
        android:hint="Введите 2 слагаемое"/>

</LinearLayout>
```

```
<Button
    android:id="@+id/tvEnter"
    android:layout_marginTop="40dp"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/tvResult"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Решить"
/>
```

```
public class MainActivity extends AppCompatActivity{

    2 usages
    TextView result;
    2 usages
    Button btnResult;
    2 usages
    EditText value1, value2;

    @Override
    3 protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        result = findViewById(R.id.tvResult);
        btnResult = findViewById(R.id.tvEnter);
        value1 = findViewById(R.id.tvValue1);
        value2 = findViewById(R.id.tvValue2);

        3 btnResult.setOnClickListener(new View.OnClickListener() { //вызываем кнопку по нажатию.
            @Override
            3 public void onClick(View v) {
                String strNum1 = value1.getText().toString().trim(); //trim() удаляет начальные и конечные пробелы
                String strNum2 = value2.getText().toString().trim();
                3 if (strNum1.equals("") || strNum2.equals("")) {
                    Toast.makeText( context: MainActivity.this, text: "Введите оба числа!", Toast.LENGTH_SHORT).show(); //Toast - всплывающие сообщения.
                    return;
                }

                result.setText("Результат: " + ( Integer.parseInt(strNum1)+Integer.parseInt(strNum2)));
            }
        });
    }
}
```

13. Реализовать мобильное приложение в Android Studio, которое перемножает три целых числа, введенных пользователем в поля ввода (EditText). Результат вычисления отобразить в лог после нажатия на кнопку (Button). В макете activity разрешено использовать контейнеры только одного типа – RelativeLayout. В приложении необходимо использовать Library Data Binding для привязки виджетов. Проверки на некорректные значения – обязательны!

ОДИН ОБЩИЙ контейнер <RelativeLayout>.

```

<TextView
    android:id="@+id/tvResult"
    android:layout_marginTop="50dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="ИТОГ"
    android:textSize="32dp"
    android:textStyle="bold"
/>

<EditText
    android:id="@+id/tvValue1"
    android:layout_marginTop="40dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:textSize="18dp"
    android:layout_below="@+id/tvResult"
    android:hint="Введите 1 число"
/>

<EditText
    android:id="@+id/tvValue2"
    android:textSize="18dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvValue1"
    android:inputType="number"
    android:hint="Введите 2 число"/>

public class MainActivity extends AppCompatActivity{

    2 usages
    TextView result;
    2 usages
    Button btnResult;
    2 usages
    EditText value1, value2, value3;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        result = findViewById(R.id.tvResult);
        btnResult = findViewById(R.id.tvEnter);
        value1 = findViewById(R.id.tvValue1);
        value2 = findViewById(R.id.tvValue2);
        value3 = findViewById(R.id.tvValue3);

        btnResult.setOnClickListener(new View.OnClickListener() { //вызываем кнопку по нажатию.
            @Override
            public void onClick(View v) {
                String strNum1 = value1.getText().toString().trim(); //trim() удаляет начальные и конечные пробелы.
                String strNum2 = value2.getText().toString().trim();
                String strNum3 = value3.getText().toString().trim();
                if (strNum1.equals("") || strNum2.equals("") || strNum3.equals("")) {
                    Toast.makeText(context: MainActivity.this, text: "Введите все числа!", Toast.LENGTH_SHORT).show(); //Toast - всплывающие сообщения.
                    return;
                }
                result.setText("Результат: " + ( Integer.parseInt(strNum1)*Integer.parseInt(strNum2)*Integer.parseInt(strNum3)));
            }
        });
    }
}

```

```

<EditText
    android:id="@+id/tvValue3"
    android:textSize="18dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvValue2"
    android:inputType="number"
    android:hint="Введите 3 число"/>

```

```

<Button
    android:id="@+id/tvEnter"
    android:layout_marginTop="40dp"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/tvValue3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Решить"
/>

```

14. Реализовать мобильное приложение в Android Studio, состоящее из двух activity, таким образом, чтобы в первую activity вернулся результат какой-то операции из второй activity. Подробнее: Вторая activity запускается по нажатию на кнопку (Button) в первой activity. Во второй activity пользователь вводит какое-то значение в EditText и нажимает на кнопку (Button). После этого приложение возвращается в первую activity и отображает в поле TextView значение, введенное пользователем.

MainActivity


```

<RelativeLayout 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">

    <TextView
        android:id="@+id/tvResult"
        android:layout_marginTop="50dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Здесь будет ваше сообщение"
        android:textSize="32dp"
        android:textStyle="bold"
    />

    <Button
        android:id="@+id/tvButton1"
        android:layout_marginTop="40dp"
        android:orientation="horizontal"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Написать данее"
    />

</RelativeLayout>

```

```

2 usages
TextView result;
2 usages
Button message;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    result = findViewById(R.id.tvResult);
    message = findViewById(R.id.tvButton1);
    Intent intent = new Intent( packageContext, this,NextActivity.class);

    message.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivityForResult(intent, requestCode: 12345); //startActivityForResult для того, чтобы указать, что мы ждём какие-то данные назад
        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) { //принимает значения из второй activity
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 12345) {
        if (data != null){
            String str = data.getStringExtra( name: "data"); //получаем данные
            Log.d( tag: "simple_app_tag", str); //записываем в лог
            result.setText(str); //устанавливаем данные в TextView
        }
    }
}

```

NextActivity

```

<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=".NextActivity">

    <EditText
        android:id="@+id/tvValue1"
        android:layout_marginTop="40dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:hint="Введите любое сообщение."
    />

    <Button
        android:id="@+id/tvButton2"
        android:layout_marginTop="40dp"
        android:orientation="horizontal"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/tvValue1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Решить"
    />

</LinearLayout>

```



```

public class NextActivity extends AppCompatActivity {

    2 usages
    EditText str;
    2 usages
    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_next);

        str = findViewById(R.id.tvValue1);
        btn = findViewById(R.id.tvButton2);
        Intent intent2 = new Intent(getApplicationContext(), MainActivity.class);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String stroka = str.getText().toString(); //забираем данные из EditText
                intent2.putExtra( name: "data", stroka); //помещаем данные в intent2
                setResult(RESULT_OK, intent2); //возвращаем intent2
                finish(); //завершаем работу activity
            }
        });
    }
}

```

15. Реализовать мобильное приложение в Android Studio, с помощью которого по нажатию кнопки (Button) можно поделиться текстом из виджета EditText через другое приложение (мессенджеры, смс и т.п.). В приложении необходимо использовать Library Data Binding для привязки виджетов.

В начале добавим поддержку библиотеки Data Binding в build.gradle(Module:app). СИНХРОНИЗУЕМ (кнопка Sync Now).

```

compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}

buildFeatures{
    viewBinding true
}

```

```

<EditText
    android:id="@+id/tvText"
    android:layout_marginTop="50dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:hint="Введите текст, которым хотите поделиться"
    android:textSize="32dp"
    android:textStyle="bold"
/>

```

```

<Button
    android:id="@+id/tvButton"
    android:layout_marginTop="40dp"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/tvText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Поделиться"
/>

```

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    4 usages
    private ActivityMainBinding binding;

    EditText str;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater()); //не забудь импортировать в build.gradle(Module:app)!
                                                                    // Здесь получаем данные из xml файла.
        setContentView(binding.getRoot()); // с помощью getRoot() мы получаем View из этого binding.

        binding.tvButton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent sendIntent = new Intent();
        String stroka = binding.tvText.getText().toString(); //принимаем данные из EditText
        sendIntent.setAction(Intent.ACTION_SEND); //загружает экран для отправки данных, указанных в номерении.
        sendIntent.putExtra(Intent.EXTRA_TEXT, stroka); //указываем тип сообщения и само сообщение
        sendIntent.setType("text/plain"); //простой текст
        startActivity(Intent.createChooser(sendIntent, title: "Поделиться")); //указываем, какое действие должно начаться и заголовок
    }
}

```

16. Реализовать мобильное приложение в Android Studio, в котором по нажатию на кнопку (Button) будет заменяться содержимое экрана, расположенного ниже кнопки (см. лекцию 2.7, слайд 38). При выполнении задания обязательно использовать фрагменты (динамическая регистрация). Содержимое экранов может быть любое.

(Запускаю TestFragmentActivity через MainActivity. Думаю, что можно и без MainActivity).
Создаём TestFragmentActivity и два фрагмента FirstFragment и SecondFragment.

```

<RelativeLayout xmlns:android="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">

    <Button
        android:id="@+id/tvButton"
        android:layout_marginTop="40dp"
        android:orientation="horizontal"
        android:layout_centerHorizontal="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Запустить FragmentActivity"
        />

</RelativeLayout>

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    1 usage
    Button tvButton = findViewById(R.id.tvButton);

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvButton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.tvButton) {
            Intent intent = new Intent( packageContext: this, TestFragmentActivity.class);
            startActivity(intent);
        }
    }
}

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.and
    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"
    tools:context=".TestFragmentActivity">

    <Button
        android:id="@+id/btnChange"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Change"
        android:textSize="38dp"
        app:layout_constraintTop_toTopOf="parent"
        />

    <RelativeLayout
        android:id="@+id/relativeContainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="38dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnChange"
        />

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

public class TestFragmentActivity extends AppCompatActivity {

    2 usages
    private int counterFragment = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_test_fragment);
        Fragment exFragment1 = new FirstFragment();
        Fragment exFragment2 = new SecondFragment();

        Button btnChange = findViewById(R.id.btnChange);
        btnChange.setOnClickListener(v -> {
            FragmentTransaction ft = getSupportFragmentManager().beginTransaction(); //получаем экземпляр Fragment
            if(counterFragment % 2 == 0){
                ft.replace(R.id.relativeContainer,exFragment1);
            }else{
                ft.replace(R.id.relativeContainer,exFragment2);
            }
            ft.commit(); //динамически загружаем фрагмент в activity
            counterFragment++;
        });
    }
}

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/a
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".FirstFragment">

    <TextView
        android:id="@+id/tvFragmentWelcome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:textSize="32sp"/>

</RelativeLayout>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SecondFragment">

    <ImageView
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerInParent="true"
        android:src="@drawable/img"
        />

</RelativeLayout>

```

```

public class FirstFragment extends Fragment {

    1 usage
    public FirstFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_first, //inflater - метод, который из содержимого
                                                                    // layout-файла может создать View-элемент.
        container,
        attachToRoot: false);
        TextView tv = rootView.findViewById(R.id.tvFragmentWelcome);
        tv.setText("It's a first fragment");
        return rootView;
    }
}

public class SecondFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_second, container, attachToRoot: false);
    }
}

```

- 17.
18. Реализовать мобильное приложение в Android Studio, в котором при нажатии на кнопку (Button) создается база данных SQLite с таблицей для хранения списка студентов (имя, фамилия, номер группы, возраст).

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

```

Добавляем разрешения в AndroidManifest!

MainActivity выглядит так:

```

2 usages
private EditText etFirstName, etLastName, etAge, etGroup;

3 usages
private DBManager dbManager; // с помощью этой переменной осуществляется доступ к базе данных и работа с ней

@SuppressLint("MissingInflatedId")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    this.dbManager = new DBManager(new MySQLiteHelper( < this, name: "my_database.db", f: null, version: 1));
    //конструктор принимает объект созданного ранее MySQLiteHelper
    etFirstName = findViewById(R.id.etFirstName);
    etLastName = findViewById(R.id.etLastName);
    etAge = findViewById(R.id.etAge);
    etGroup = findViewById(R.id.etGroup);

    Button btnAdd = findViewById(R.id.btnAdd);
    Button btnContinue = findViewById(R.id.btnContinue);
    btnAdd.setOnClickListener(this);
    btnContinue.setOnClickListener(this);
}

Override
public void onClick(View v) {
    if(v.getId() == R.id.btnAdd){ // если нажали на кнопку "Add"
        if(this.dbManager != null){
            String fName = etFirstName.getText().toString();
            String lName = etLastName.getText().toString();
            String age = etAge.getText().toString();
            String group = etGroup.getText().toString();
            if(!fName.isEmpty() && !lName.isEmpty() && !age.isEmpty() && !group.isEmpty()){
                boolean result = dbManager.savePersonToDataBase(new Person(fName,lName,Integer.parseInt(age),group));
                if(result){
                    Toast.makeText( context: this, text: "Запись успешно добавлен",Toast.LENGTH_LONG).show(); //лучше записывать в string
                }else{
                    Toast.makeText( context: this, text: "При добавлении произошла ошибка",Toast.LENGTH_LONG).show();
                }
            }else{
                Toast.makeText( context: this, text: "Вы ввели некорректное значение",Toast.LENGTH_LONG).show();
            }
        }
    }else if(v.getId() == R.id.btnContinue){ //если нажали на кнопку "Continue"
        startActivity(new Intent( packageContext: this, PersonActivity.class));
    }
}

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginTop="32dp">

    <EditText
        android:id="@+id/etFirstName"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:paddingStart="6dp"
        android:paddingEnd="6dp"
        android:hint="name"
    />

    <EditText
        android:id="@+id/etLastName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:gravity="center"
        android:layout_marginTop="6dp"
        android:paddingStart="6dp"
        android:paddingEnd="6dp"
        android:hint="lname"
    />

```

Здесь не всё, ещё 2 EditText и 2 Button.

Класс (не Activity) DBManager:

```

private SQLiteOpenHelper sqLiteHelper;

1 usage
public DBManager(SQLiteOpenHelper sqLiteHelper) { this.sqliteHelper = sqLiteHelper; }

1 usage
public boolean savePersonToDataBase(Person person){
    SQLiteDatabase db = this.sqliteHelper.getWritableDatabase(); //получили ссылку на базу данных
    //создание переменной ключ-значение по сути формируем запись для вставки в БД
    ContentValues cv = new ContentValues();
    cv.put("first_name",person.getFirstName());
    cv.put("last_name",person.getLastName());
    cv.put("age",person.getAge());
    cv.put("group", person.getGroup());
    long rowId = db.insert( table: "TABLE_PERSONS", nullColumnHack: null, cv); //метод возвращает номер строки в
    // случае успешной вставки или -1 в случае ошлбк

    cv.clear(); //очистка
    db.close(); // закрытие базы данных
    return rowId != -1; //возвращаем результат вставки
}

```

```

public ArrayList<Person> loadPersonFromDatabase(){ // метод для выборки из таблицы всех записей
    ArrayList<Person> persons = new ArrayList<>();
    SQLiteDatabase db = this.sqliteHelper.getWritableDatabase();
    Cursor dbCursor = db.query( table: "TABLE_PERSON",
        columns: null, selection: null, selectionArgs: null,
        groupBy: null, having: null, orderBy: null);

    if (dbCursor.moveToFirst()){
        do{
            String fName = dbCursor.getString(dbCursor.getColumnIndexOrThrow( columnName: "first_name"));
            String lName = dbCursor.getString(dbCursor.getColumnIndexOrThrow( columnName: "last_name"));
            int age = dbCursor.getInt(dbCursor.getColumnIndexOrThrow( columnName: "age"));
            String group = dbCursor.getString(dbCursor.getColumnIndexOrThrow( columnName: "group"));
            persons.add(new Person(fName, lName, age, group));
        }while (dbCursor.moveToNext());
    }

    dbCursor.close();
    db.close();
    return persons;
}

```

Класс Person:

```

3 usages
private String firstName;

3 usages
private String lastName;

3 usages
private int age;

3 usages
private String group;

2 usages
public Person(String firstName, String lastName, int age, String group){
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.group = group;
}

1 usage
public String getFirstName() { return firstName; }

public void setFirstName(String firstName) { this.firstName = firstName; }

1 usage
public String getGroup(){return group;}

public void setGroup(String group){this.group = group;}

1 usage
public String getLastName() { return lastName; }

public void setLastName(String lastName) { this.lastName = lastName; }

1 usage
public int getAge() { return age; }

```

Класс MyAppSQLiteHelper

```

1 usage
public class MyAppSQLiteHelper extends SQLiteOpenHelper {

    1 usage
    public MyAppSQLiteHelper(Context c, String name, SQLiteDatabase.CursorFactory f, int version){
        super(c, name, f, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + "TABLE_PERSONS" + "(" // создание таблицы, которая будет храниты
            + "_id integer primary key autoincrement, " //присваиваем ключ-идентификатор
            + "first_name text,"
            + "last_name text,"
            + "age integer,"
            + "group text" + ");"
        );
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) { // вызывается при модифи
    }
}

```

19.

20. Реализовать мобильное приложение в Android Studio, в котором значение, записанное в поле виджета EditText, сохраняется в файл во внутренней памяти устройства при нажатии на кнопку (Button). Имя файла задает пользователь (вводит во второй EditText).

```
<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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/etText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Введите текст"/>

    <EditText
        android:id="@+id/etFileName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Введите имя файла"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Сохранить"/>

</LinearLayout>

public class MainActivity extends AppCompatActivity {
    2 usages
    EditText editText1, editText2;
    2 usages
    Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText1 = findViewById(R.id.etText);
        editText2 = findViewById(R.id.etFileName);
        button = findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String text = editText1.getText().toString();
                String fileName = editText2.getText().toString();
                FileOutputStream outputStream;

                try {
                    outputStream = openFileOutput(fileName, Context.MODE_PRIVATE);
                    outputStream.write(text.getBytes());
                    outputStream.close();
                    Toast.makeText(getApplicationContext(), text + "Файл сохранен", Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

21. Реализовать мобильное приложение в Android Studio, в котором по нажатию на первую кнопку (Button) будет запускаться фоновый сервис. По нажатию на вторую кнопку сервис должен быть остановлен. Вызовы методов жизненного цикла activity и сервиса должны логироваться.

```
<user-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

//Киллометровый код

22. Реализовать мобильное приложение в Android Studio, которое при смене языка на устройстве выводит всплывающее сообщение (Toast) на экран и пишет эту информацию в лог (или в текстовый файл на устройстве).

