

**Министерство образования Пензенской области**  
**Отдел образования Кузнецкого района**

**Тема:**

**"Взаимодействие с сервером мобильного приложения"**

**Игра "поймай покемона"**

/секция – информатика и ИКТ/

Выполнил работу:

Лобанов Рамиль

ученик 10 класса

МБОУ СОШ с. Махалино

Руководитель работы:

Тенишев Марат Дамирович

учитель информатики и ИКТ

МБОУ СОШ с. Махалино

Телефон: (841-57) 55-5-92

Адрес: 442519, Пензенская

область, Кузнецкий район, с.

Махалино, ул. Рабочий городок

совхоза, д. 16

Телефон (841-57)55-1-06

2020

## Оглавление

1. Введение .....	3
2. Выбор среды разработки.....	5
2.1. Необходимые ресурсы для проекта.....	5
3. Основная часть.....	6
3.1. Разработка приложения.....	6
3.1.1. Класс MyTask.....	6
3.1.2. Класс Send.....	7
3.1.3. Класс TwoG.....	8
3.1.4. Класс oneG.....	10
3.1.5. Сервер PHP.....	11
4. Заключение.....	12
1. Источники информации.....	13

## **1. Введение**

### Проблема.

В современном мире Интернет стал не только хранителем информации, но и позволил нам автоматизировать управление многими аспектами реальной жизни. Для этого необходимо чтобы наши устройства и приложения взаимодействовали с серверами Интернета.

### Актуальность.

В эпоху "промышленной революции 4.0" в нашу повседневную жизнь внедряются такие современные услуги как "Умный дом", "Умный город", "Электронное правительство" и т.п., поэтому мобильным приложениям необходимо постоянно обмениваться информацией с серверами Интернета.

### Гипотеза.

При помощи базовых знаний PHP и Java можно организовать обмен данными Android-приложения с сервером.

### Цель.

Создать приложение для обмена данными с сервером Интернета.

### Задачи.

1. Выбрать среду разработки.
2. Разработать приложение-игру для мобильных устройств с ОС Android.
3. Настроить сервер для обмена данными с приложением.
4. Определить направления по дальнейшему развитию проекта.

### Методы исследования.

1. Моделирование.
2. Компьютерный эксперимент.

### Предмет исследования.

Предметом исследования являются инструменты Android и PHP для обмена данными с сервером (рис.1).

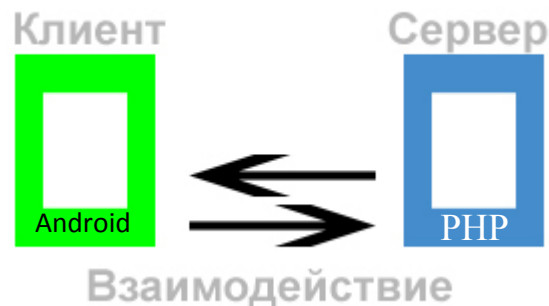


Рис.1

## 2. Выбор среды разработки.

### 2.1. Необходимые ресурсы для проекта.

Для решения поставленной задачи мы выберем Eclipse, так как он гораздо менее требователен к ресурсам ПК, чем Android Studio и относится к свободному программному обеспечению.

Для работы с Eclipse необходимы инструменты: – Java SE Development Kit 8, Android Development Tools (ADT) и SDK Manager. Все они собраны в дистрибутиве `adt-bundle-windows-x86-20131030` (рис.2).

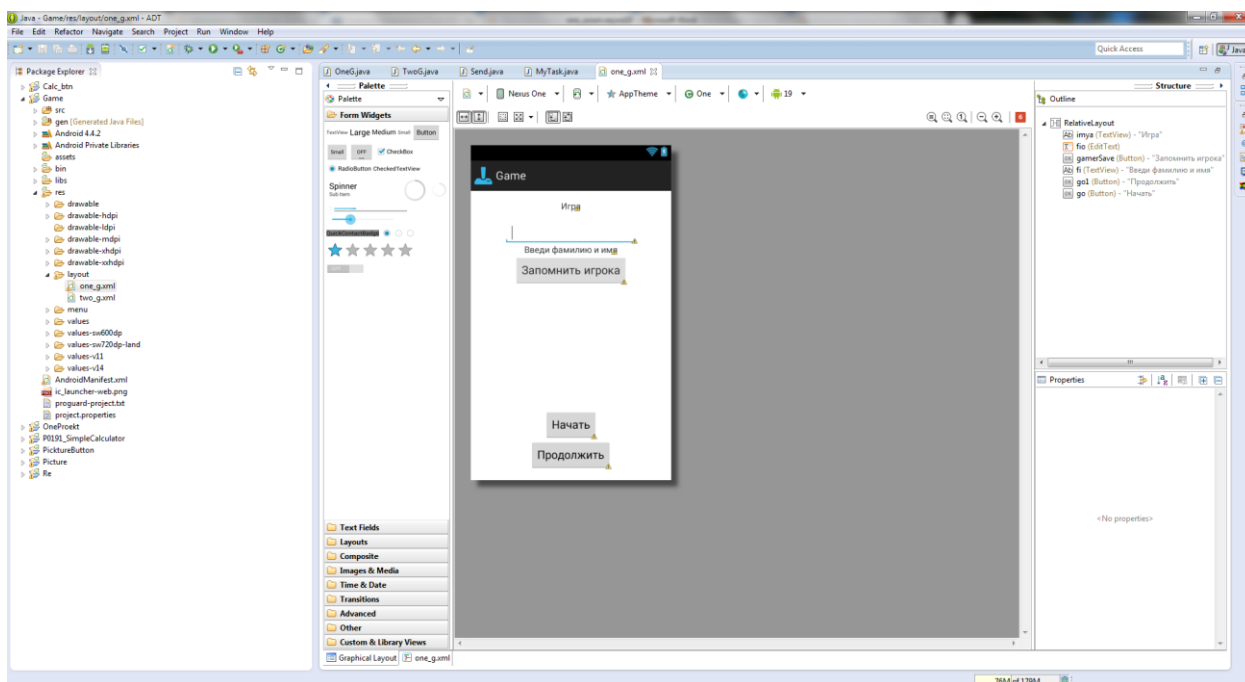


Рис.2

**Вывод:** при помощи Eclipse можно создать мобильное приложение для OS Android, которое будет взаимодействовать с сервером.

### 3. Основная часть.

#### 3.1. Разработка приложения.

##### 3.1.1. Класс MyTask.

Расширим класс MyTask за чет абстрактного класса AsyncTask для решения асинхронных задач игры:

- генерирование случайным образом координат изображений "покемонов" и вывод их через каждые 2 секунды на экран;
- запись и показ промежуточных результатов игры;
- запись и показ окончательного результата игры.

```
public class MyTask extends AsyncTask<Void,Integer,Void> {
    protected void onPreExecute(){
        TwoG.Info.setText("Старт игры");
    }
    /*переопределить метод onProgressUpdate для публикации промежуточных обновлений в
    интерфейс игры*/
    protected void onProgressUpdate(Integer...values){
        TwoG.Image.setX(values[0]);
        TwoG.Image.setY(values[1]);
        TwoG.Image.setImageResource(TwoG.pic[values[2]]);
        TwoG.Image.setTag(values[2]);
        TwoG.Info.setText("");
        TwoG.Lay.setBackgroundColor(Color.parseColor("#00ffffff"));
        if(values[3]<19){
            TwoG.Info.setText("Правильных-"+TwoG.yes+" Неправильных-
            "+TwoG.no);}
        if(values[3]==19)
            TwoG.Info.setText("Игра окончена\n"+"Правильных-
            "+TwoG.yes+"Неправильных-"+TwoG.no);
        }
    /*основной метод для 20 показов 8 картинок покемонов из массива pic,
    зарегистрированных в классе dravable файла R и отображения на тестовой наклейке info
    промежуточных результатов игры, который выполняется в отдельном потоке*/
    protected Void doInBackground(Void... voids) {
        try{
            for(int i=0;i<20;i++){
                TimeUnit.MILLISECONDS.sleep(2000);
                int x=(int)(Math.random()*(TwoG.width-150))+50;
                int y=(int)(Math.random()*(TwoG.height-150))+50;
                int z=(int)(Math.random()*7);
                /*publishProgress передает значения полей в onProgressUpdate
                (x, y-случайные координаты картинки;
                z-случайная картинка из массива;
                i-количество появления картинок)*/
                publishProgress(x,y,z,i);
            }
        }
        catch(InterruptedException e){}
        return null;
    }
}
```

**Вывод:** класс MyTask работает в дополнительном потоке.

### 3.1.2. Класс Send.

Расширим класс Send за счет абстрактного класса AsyncTask для решения асинхронных задач взаимодействия приложения с сервером:

- Подключиться к серверу  
(...<http://leninoschool.edu-penza.ru/marat/hand.php?new=...>);
- создать файл на сервере;  
(...[TwoG.file...](#));
- записать окончательный результат игры  
(...[URLLEncoder.encode\(TwoG.send, "UTF-8"\)...](#));
- открыть входящий поток, прочитать с сервера информацию в строку  

```
InputStreamReader reader = new InputStreamReader(connection.getInputStream());
    BufferedReader in= new BufferedReader(reader);
    str=in.readLine().toString();
    }
```
- вывести строку на текстовую наклейку  

```
TwoG.Info.setText("Записано; "+str);
```

**Вывод:** класс Send работает в дополнительном потоке.

### 3.1.3. Класс TwoG

Создадим основной класс для игры TwoG:

- Организуем прием фамилии игрока с активности OneG  
`MyBundle = getIntent().getExtras();` //организовать прием посылки
- Программно разместим виджеты на разметке

```
/*создать кнопку для отправки результата игры на сервер*/
display=getWindowManager().getDefaultDisplay();
Point size=new Point();
display.getSize(size);
width=size.x;
height=size.y;
Image=new ImageView(this);
Lay.addView(Image);

/*создать кнопку для отправки результата игры на сервер*/
Button save=new Button(this);
save.setText("save");
save.setX(340);
save.setY(575);
Lay.addView(save);

/*создать кнопку для перехода на первую активность*/
Button back=new Button(this); //создать экземпляр класса "кнопка"
back.setText("назад"); //создать надпись
back.setX(0); //установить координату кнопки по оси X
back.setY(575); //установить координату кнопки по оси Y
Lay.addView(back); //установить кнопку на разметку

/*создать текстовую наклейку*/
Info=new TextView(this);
Lay.addView(Info);
```

- Создадим обработчики:

```
/*создать обработчик касаний экрана*/
Lay.setOnTouchListener(new OnTouchListener(){
public boolean onTouch(View view, MotionEvent event){
if(event.getAction()==MotionEvent.ACTION_DOWN){
if((Math.abs(event.getX()-Image.getX())<100 &&
Math.abs(event.getY()-Image.getY())<100)
&&(Image.getTag().equals(0) || Image.getTag().equals(1)||
Image.getTag().equals(2)|| Image.getTag().equals(3))){
Lay.setBackgroundColor(Color.argb(200, 200, 255, 200));
yes++;} //подсчет касаний правильных картинок
else{
Lay.setBackgroundColor(Color.argb(200, 255, 200, 200));
no++;} //подсчет касаний неправильных картинок
}
}
return true;
}});
```



```

/*создать обработчик нажатия на кнопку для отправки результата игры на сервер*/
save.setOnClickListener(new OnClickListener(){
    @ Override
    public void onClick (View view){
        file="ramil.txt";//имя файла
        operation=1;// 1-создать
        send=" "+MyBundle.getString("f")+ " попал "+yes+" раз;
            ошибся "+no+" раз";//переданное значение
        write();//вызвать метод "write" для записи данных на сервер
    });

/*создать обработчик нажатия на кнопку для перехода на первую
активность*/
back.setOnClickListener(new OnClickListener(){
    @ Override
    public void onClick (View view){
        startActivity(it);
        finish();
    });
}

```

- Создадим асинхронные потоки:

```

/*создать поток для управления процессом игры*/
MyTask Number=new MyTask();//создать новый поток
Number.execute();//запустить поток

/*создать поток для записи результата на сервер*/
public static void write(){Send newSend = new Send();
newSend.execute();}

```

**Вывод:** Приложение выводит картинки, ведет счет игры, взаимодействует с сервером.

### 3.1.4. Класс oneG.

Данный класс связан с первоначальной разметкой экрана приложения и предназначен для передачи данных игрока на активность TwoG.

```
public void newFioSet (View vi){
    nik=fio.getText().toString();
    fi.setText("Начнем "+nik+"?");
}

public void go (View vi){
    Intent intent=new Intent(this,TwoG.class);
    nik=fio.getText().toString();// получить фио игрока в строковую переменную
    intent.putExtra("f", nik);//упаковать посылку
    startActivity(intent);
    finish();
}
```

**Вывод:** класс oneG получает данные игрока и передает "посылку" классу TwoG.

### 3.1.5. Сервер PHP.

Напишем сервер на php и разместим в Интернете. Приложение использует get-запрос. Если на стороне клиента было указано имя файла и отправлена "1", то на стороне сервера создается файл с указанным именем и в него записывается информация отправленная клиентом. Возвращается "make"

```
<?php
$n=$_GET["new"];
$name=$_GET['name'],
$a=$_GET['get'];
if ($name!="")
{
  if ($n==1)
  {
    $fp = fopen("i/".$name, "w");
    fclose($fp);
    echo json_encode("make");
  }
}
?>
```

**Вывод:** сервер отвечает на запросы клиента.

#### **4. Заключение.**

##### Вывод:

В ходе выполнения проекта мне удалось создать приложение для обмена данными с сервером Интернета.

Цель проекта достигнута – создано приложение для ОС Android, которое успешно обменивается данными с сервером Интернета.

##### Развитие проекта:

- Обновить дизайн приложения-клиента.
- Дописать серверное приложение и клиент для ведения истории игрока.

## **1. Источники информации.**

1. И.Б. Емельянов (Р.И. Емельянов). Учебно-методическое пособие "Компьютерные науки: основы веб-разработки и клиент-серверные технологии". Часть 3 Пенза, 2020.
2. И.Б. Емельянов (Р.И. Емельянов). Учебно-методическое пособие "Компьютерные науки: изучение языка программирования Java". Часть 3 Пенза: Министерство образования Пензенской области. ИРР Пензенской области, 2020.