

# Курсов проект по мобилни приложения

Тема:

*“Забележителностите в квартал Лозенец”*

Изготвил:

*Име: Цветислав Викторов*

*Специалност: Компютърни науки*

*ФН: 80842*



# ANDROID

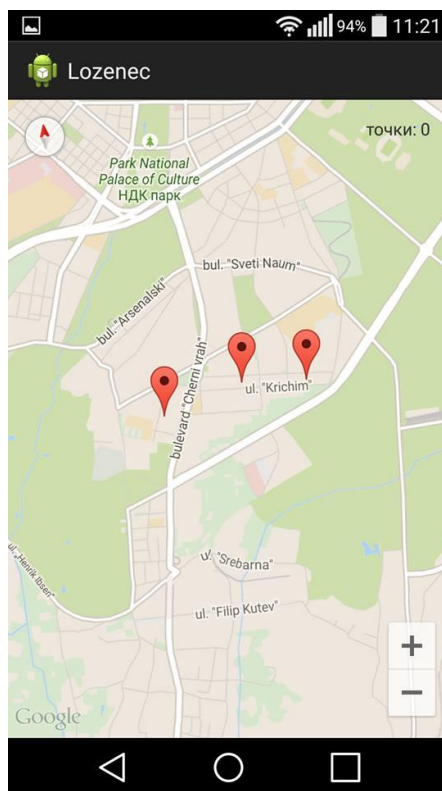
**Lozenec** е приложение за Андроид, чиято цел е да запознае хората, които го използват, със забележителностите в региона. А именно когато се намират в квартал Лозенец, имат възможност да отговарят на въпроси, свързани с исторически и научни интереси. То проверява до колко са запознати с фактите, намиращи се в квартала. Потребителите могат да отговарят на въпроси само ако са на по – малко от 40 метра до съответната забележителност.

Приложението се състои от четири класа:

- [MAINACTIVITY.JAVA](#)
- [QUESTION.JAVA](#)
- [QUESTIONACTIVITY.JAVA](#)
- [QUESTIONBANK.JAVA](#)

## MainActivity.java

Съдържа карта, върху която се появяват “Маркери”. Тези “Маркери” когато бъдат натиснати от потребителя, се появява съответния въпрос. Потребителя има достъп до не повече от 3 въпроса.



С фрагмента се задава местоположението на “маркерите” на картата за съответните въпроси. Видими са най – много 3 “маркера”:

```
for (int i = 0; i < numOfQuestions; i++) {
    Marker m = map.addMarker(new MarkerOptions()
        .position(QuestionBank.questions.get(i).position)
        .snippet(i+"")
        .visible(false));
    markers[i]=m;
    if (i<visible) {
        m.setVisible(true);
    }
}
```

Методът **private void** loadQuestions() зарежда въпросите, които се намират в текстовия файл “questions.txt”. Там те са предварително зададени от администратора.

Методът **public boolean** MarkerClick(Marker marker) се използва за да създаде събитие, което ще бъде извършено при натискане на съответния “Маркер”. При натискане, на потребителя е изписан съответния въпрос, отговарящ на съответната забележителност. Прави се проверка за нужното разстояние между обекта и потребителя и ако го удовлетворява се появява на екрана.

```
public boolean onMarkerClick(Marker marker) {
    Intent i = new Intent(this, QuestionActivity.class);
    String quesNum = marker.getSnippet();
    int num = Integer.parseInt(quesNum);

    Location questionLocation = new Location("name");
    questionLocation.setLatitude(QuestionBank.questions.get(num).
        position.latitude);
    questionLocation.setLongitude(QuestionBank.questions.get(num).
        position.longitude);
    double distance=currentLocation.distanceTo(questionLocation);

    if(distance<MINIMUM_DISTANCE){
        i.putExtra("questionNumber", num);
        startActivityForResult(i, num);
    }
    return false;
}
```

Методът **protected void** onActivityResult(int requestCode, int resultCode, Intent data) съобщава на потребителя дали е отговорил правилно или не. Ако е правилен отговора се добавя точка към общия му резултат, изчезва текущият въпрос и се добавя следващ. Ако не е – появява му се съобщение за грешка, въпросът изчезва и не се появява следващ въпрос.

```
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK) {
        if(data.getBooleanExtra("correct", false)){
            score++;
        }
    }
}
```

```

        tv.setText("Точки: "+String.valueOf(score));
        markers[requestCode].setVisible(false);
        if(visible<numOfQuestions){
            markers[visible].setVisible(true);
            visible++;
            Toast t1=Toast.makeText(this, "Верен отговор",
            Toast.LENGTH_LONG);
            t1.show();
        }

        else{
            Toast t2=Toast.makeText(this, "Грешен отговор",
            Toast.LENGTH_LONG);
            t2.show();
            markers[requestCode].setVisible(false);
        }
    }
}

```

## Question.java

Този клас определя обекта “въпрос”. Всеки “въпрос” си има текст, четири възможни отговора, номер на верен отговор и координатите му на картата.

```

Private String qText;
Private String[] answers = new String[4];
Private int correctAns;
Private LatLng position;

```

## QuestionBank.java

Обектът от този тип представлява един статичен публично достъпен списък с въпросите от тип Question.

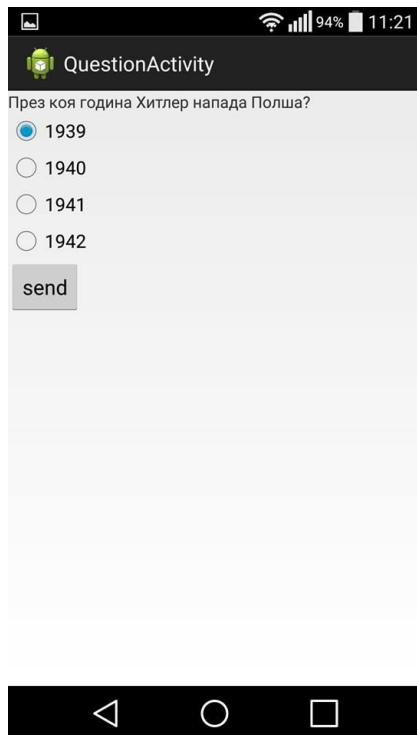
```

static public ArrayList<Question> questions = new ArrayList<Question>();

```

## QuestionActicity.java

Този клас е активитито за въпроса, който се появява когато потребителя е натиснал “Маркер” за съответната забележителност и в него се появява съответния въпрос, с възможните четири отговора. Има четири радио-бутона всеки съдържащ текст с възможен отговор. Когато потребителя си избере отговор с радио - бутона натиска бутона “send”.



При натискане на бутона се извиква метода по –  
надолу и той проверява за коректността на  
отговора. Според него се задава Intent, който  
съдържа булева стойност за вярност.

```
public void onClick(View v) {
    if(isCorrect()){
        Intent i = new Intent();
        i.putExtra("correct", true);
        setResult(RESULT_OK,i);
        finish();
    }
    else{
        Intent i = new Intent();
        i.putExtra("correct", false);
        setResult(RESULT_OK,i);
        finish();
    }
}
```

Самият метод `isCorrect()` проверява за обекта от тип `Question` дали  
отговорения вариант съвпада със зададения верен отговор от текстовия  
файл "Question.txt"

```
Boolean isCorrect(){
    int marked=-1;
    if (r1.isChecked()) {
        marked=1;
    }
    if (r2.isChecked()) {
        marked=2;
    }
    if (r3.isChecked()) {
        marked=3;
    }
    if (r4.isChecked()) {
        marked=4;
    }
    if (marked == q.correctAns) {
        return true;
    }
    return false;
}
```