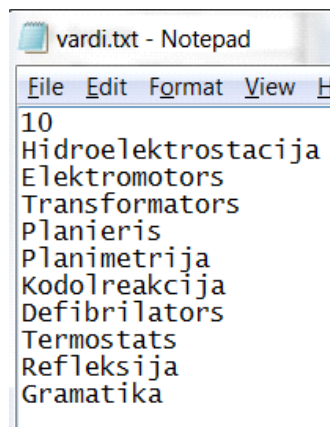


## DARBS AR TEKSTA DATNĒM

### 21.nodarbība - uzdevumi

#### 1. UZDEVUMS

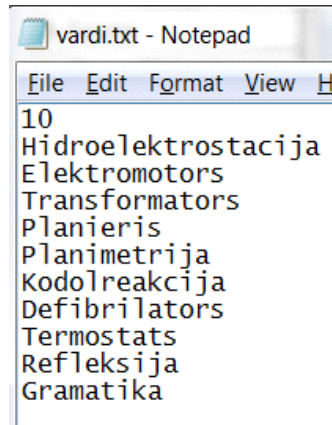
1. Dota datne vardi.txt un programma, kas uz ekrāna parāda vienu no šīs datnes vārdiem. Kuru vārdu parādīs šī programma?



```
public void paradiVardu(View v){
    EditText vards=(EditText)findViewById(R.id.editText1);
    try{
        BufferedReader lasa= new BufferedReader(new InputStreamReader(getAssets().open("vardi.txt")));
        String rinda=lasa.readLine();
        rinda = lasa.readLine();
        vards.setText(rinda);
        lasa.close();
    }catch(Exception e){
        vards.setText("lasot slikti");
    }
}
```

- a) Hidroelektrostacija
- b) Elektromotors
- c) Gramatika

2. Dota datne vardi.txt un programma, kas uz ekrāna parāda vienu no šīs datnes vārdiem. Kuru vārdu parādīs šī programma?



```
public void paradiVardu(View v){
    EditText vards=(EditText)findViewById(R.id.editText1);
    try{
        BufferedReader lasa= new BufferedReader(new InputStreamReader(getAssets().open("vardi.txt")));
        String rinda=lasa.readLine();
        int n=Integer.parseInt(rinda);
        for(int i=0;i<n;i++){
            rinda = lasa.readLine();
        }
        vards.setText(rinda);
        lasa.close();
    }catch(Exception e){
        vards.setText("lasot slikti");
    }
}
```

- a) Hidroelektrostacija
- b) Elektromotors
- c) Gramatika

3. Dota datne vardi.dat, kura jau iepriekš programmā ir papildināta ar vārdiem. Izpildās programmas metode uzDatni. Kā izskatīsies datne vardi.dat?



```
public void uzDatni(View v){
    EditText editText1=(EditText)findViewById(R.id.editText1);
    try {
        PrintWriter printWriter=new PrintWriter(new FileWriter(getFilesDir()+File.separator+"vardi.dat", true));

        printWriter.println("Kailcirte");

        printWriter.close();
    } catch (Exception e) {
        editText1.setText("nav faila?");
    }
}
```

a)



b)



c)



4. Dota datne vardi.dat, kura jau iepriekš programmā ir papildināta ar vārdiem. Izpildās programmas metode uzDatni. Kā izskatīsies datne vardi.dat?



```
public void uzDatni(View v) {
    EditText editText1=(EditText)findViewById(R.id.editText1);
    try {
        PrintWriter printWriter=new PrintWriter(new FileWriter(getFilesDir()+File.separator+"vardi.dat"));

        printWriter.println("Kailcirte");

        printWriter.close();
    } catch (Exception e) {
        editText1.setText("nav faila?");
    }
}
```

a)



b)



c)



## UZDEVUMS PATSTĀVĪGAI PROGRAMMĒŠANAI

Papildināt zoles pieraksta programmu ar iespēju datnē saglabāt arī spēlētāju vārdus un vēlāk tos nolasīt un parādīt kopā ar punktu summu.

Iespējamie risinājumi var būt vairāki, tiek piedāvāts šāds plāns:

- 1) Papildinām projektu ar vēl 3 teksta lodziņiem, kuros varēs ievadīt spēlētāju vārdus.
- 2) Kad pirmo reizi notiek punktu saglabāšana datnē, kā pirmās 3 rindas saglabājam spēlētāju vārdus.
- 3) Turpmāk vārdus glabāt vairs nevajag, taču nolasot jāatceras, ka pirmās 3 rindas jānolasa atsevišķi, nevis lielajā punktu summēšanas ciklā.
- 4) Lai piefiksētu, ka glabāšana vienreiz jau ir notikusi un vairāk vārdus glabāt nevajag, atkal var rīkoties dažādi:
  - Priekšlikums ir – ieviešam *boolean* mainīgo *vardi*. Kad glabājam datnē, pārbaudām – ja *vardi* ir *true*, vārdus datnē nav jāglabā, pretējā gadījumā (pirmajā reizē) uzliekam *vardi=true*.
  - Cits variants būtu pievienot speciālu vārdu glabāšanas pogu, kas pēc pirmās nospiešanas paliek neaktīva.
  - Gan šim, gan pirmajam variantam ir arī savi trūkumi – neviens no tiem nekontrolē, vai vārdi lodziņos ir ierakstīti, tāpat arī punkti.

Lai izdodas!