

TALSU VALSTS ĢIMNĀZIJA

Automatizēta dzīves apraksta izveide

Zinātniski pētnieciskais darbs dabaszinātņu nozarē

Talsu Valsts ģimnāzijas
11. klases skolniece
Jana Ozolniece

Darba vadītājs: Krišjānis Roze

Anotācija

Automatizēta dzīves apraksta izveide. Jana Ozolniece, darba vadītājs Talsu Valsts ģimnāzijas programmēšanas skolotājs Krišjānis Roze.

Pētījums tika veikts, izstrādājot programmu.

Darba mērķis: izveidot digitālu rīku, kas palīdzētu cilvēkiem ātrāk un efektīvāk atrast darbu, izmantojot automatizētas CV izveides tehnoloģijas un algoritmus.

Pētījums tika veikts programmatūras izstrādes vidē Eclipse, valodā java. Galvenais rezultāts - programma izveido CV docx formātā.

Pētījumā secināts, ka ir pieejami neskaitāmi daudz CV paraugu pēc kuriem vadīties. Automatizēta dzīves apraksta izveide atvieglo CV sastādīšanas darbu, ko bez šī pētījuma rezultāta darba meklētājam būtu jāveic.

Atslēgas vārdi : CV, java, programmēšana.

Annotation

Automated resume creation. Jana Ozolniece, head of work, Talsi State Gymnasium programming teacher Krišjānis Roze.

The research was carried out while developing the program.

The goal of the work: to create a digital tool that would help individuals find work faster and more efficiently using automated resume creation technologies and algorithms.

The research was carried out in the software development environment Eclipse, in java language. The main result - the program creates a CV in docx format.

The research concluded that there are countless CV samples available to guide you. Automated resume creation eases the job of creating a resume that a job seeker would have to do without the result of this research.

Keywords: resume, java, programming.

Saturs

IEVADS.....	4
1.TEORĒTISKĀ DAĻA	5
1.1 Programmēšanas valoda - Java	5
1.2 CV	5
1.3 Apache POI.....	5
1.4 Eclipse IDE.....	6
1.5. Kopsavilkums	6
2.PRAKTISKĀ DAĻA.....	7
2.1 Bibliotēku izvēle.....	7
2.2 CV jautājumu izveide	7
2.3 Datu saglabāšana.....	7
2.4 Word dokumenta izveide.....	8
2.5 Datu ievietošana word dokumentā	8
3.REZULTĀTU ANALĪZE.....	10
SECINĀJUMI.....	11
LITERATŪRAS SARAKSTS	12

Ievads

Zinātniski pētnieciskā darba tēma - “Automatizēta dzīves apraksta izveide”, tika izvēlēta, lai atvieglotu dzīves apraksta, jeb CV izveidi. “Automatizēta dzīves apraksta izveide” ir aktuāls rīks, jo arvien lielāka iedzīvotāju daļa jūtas nedroši veidojot savus dzīves aprakstus, jeb CV.

Mūsdienās arvien aktuālāka tēma kļūst elektronisks CV darba pieteikums docx formātā. Pieejamie interneta līdzekļi ir daudz un dažādi, taču tiem ir problēmas, vai nu tie nav viegli pieejami, vai nu dotie jautājumi ir neskaidri, vai nu CV izveidošana ir maksas pakalpojums. Meklējot CV svešvalodā, kā piemēram, angļu valodā, rezultātu ir vairāk, bet, ja lietotājam nav labas svešvalodu zināšanas, daži jautājumi var šķist mulsinoši. Programma, kas ļauj izveidot savu dzīves aprakstu, uzdodot vieglus jautājumus, sniedz iespēju viegli, saprotamā veidā, izveidot pašam savu CV.

Izpētot internetā piedāvātos rīkus, autoram radās interese izpētīt, kā izveidot programmu, kas izveidotu word dokumentu. Izpētot pieejamos paraugus, bija svarīgi noformulēt CV prasīto, lietotājiem saprotamākā veidā.

Darba tēma. Automatizēta dzīves apraksta izveide.

Darba mērķis. Radīt programmu, kas atvieglo CV izveidi.

Darba uzdevumi.

1. CV parauga izvēle.
2. Bibliotēku izvērtēšana.
3. Programmas izveide.
4. Atklādošana.

Pētījuma jautājums. Kā izveidot programmu, kas rada word dokumentu ar lietotāja datiem?

Darbā izmantotās metodes. Programmas izveide, literatūras analīze.

1. Teorētiskā daļa

1.1 Programmēšanas valoda - Java

Java ir augsta līmeņa uz klasēm balstīta objektorientēta programmēšanas valoda, kas izstrādāta, kas prasītu pēc iespējas mazākus datora atmiņas resursus. Tā ir vispārēja lietojuma programmēšanas valoda, kas paredzēta programmētājiem, lai rakstītu un apkopotu Java kodus. [1]

Java kods var darboties visās platformās, kas atbalsta Java, bez nepieciešamības atkārtoti kompilēt. Tas nozīmē, ka tas var darboties vairākās platformās. Java lietojumprogrammas parasti tiek kompilētas baitkodā, kas var darboties jebkurā Java virtuālajā mašīnā neatkarīgi no pamatā esošās arhitektūras. [1]

Java sintakse ir līdzīga programmēšanas valodai C un C++, taču tai ir mazāk zema līmeņa funkciju. *Zema līmeņa programmēšanas valoda ir programmēšanas valoda, kas nav atkarīga no datora arhitektūras.*[1]

1.2 CV

CV, vai curriculum vitae ir īss pārskats par personas dzīves gājumu - izglītību, kvalifikāciju un iepriekšējām profesijām. Izmantojot interneta resursus, ir iespējams atrast vairākus CV paraugus pēc kuriem vadīties. Izlīdzināt tekstu šajā un nāk. rindk.!

Autors izvēlējās paraugu no vietnes[2] , lai izpētītu pilnīgam dzīvesgājuma aprakstam nepieciešamās detaļas. Izpētot citus CV, tika lemts pievienot telefona numuru. Izpētot dažādus internetā pieejamos CV paraugus[3], darba autors nolēma CV iekļaut šādu informāciju:

- vārds un uzvārds;
- dzīves vietas adrese;
- telefona numurs;
- e-pasta adrese;
- vakance uz kuru kandidē vai joma, kurā tiek meklēts darbs;
- iegūtā izglītība;
- pieredze;
- profesionālie sasniegumi;
- papildus izglītība;
- svešvalodu prasme.

Docx formāts ir dokumenta formāts, ko izmanto Microsoft Word 2007 un jaunākas versijās. Tas ir Office Open XML (OOXML) formāts, kas ir atvērtā standarta formāts, ko izstrādāja Microsoft Corporation. Docx formāts ir paredzēts, lai saglabātu teksta dokumentus, tādējādi tas var saturēt tekstu, tabulas, attēlus, un citus elementus, kā arī dažādus stilus un formatējumus.

1.3 Apache POI

Apache POI ir atvērtā koda bibliotēka, ko ir izstrādājusi Apache Software Foundation, kas ļauj Java programmēšanas valodā izveidot un apstrādāt Microsoft Office dokumentus, piemēram, Word, Excel un PowerPoint, kā arī programmas, kas izmanto *MFC Microsoft Foundation Class Library* rekvizītu kopas.[4]

POI bibliotēka ir plaši izmantota automatizētos datu apstrādes projektos, jo tā ļauj automatizēti izveidot un apstrādāt Office dokumentus, bez nepieciešamības izmantot Microsoft Office programmu.

XWPF ir Apache POI bibliotēkas daļa, kas ļauj Java programmēšanas valodā izveidot un apstrādāt Microsoft Word dokumentus. XWPF ir atbilstošs Office Open XML (OOXML) formātam, kuru izmanto Microsoft Word 2007 un jaunākās versijās. [5]

1.4 Eclipse IDE

Eclipse IDE (Integrated Development Environment) ir atvērtā koda programmēšanas vides platforma, kas ļauj programmēt dažādās valodās, tostarp Java, C++, Python, un citās. Tā ir viena no vispopulārākajām bezmaksas programmēšanas vidēm, un tā ir izstrādāta Eclipse Foundation.[11]

Eclipse IDE nodrošina daudzas funkcijas, kas palīdz programmētājam veikt programmēšanas uzdevumus, piemēram:

- sintakses un semantikas pārbaude. Auto-completion un koda tips(Auto-completion (angļu valodā: automātiskā pabeigšana) ir programmēšanas vides funkcija, kas automātiski piedāvā un ievada koda fragmentus, pamatojoties uz lietotāja ievadīto tekstu un kontekstu. Tas var tikt izmantots, lai palīdzētu ātrāk un precīzāk rakstīt kodu, samazinot kļūdas un paātrinot programmēšanas procesu.);
- projektu un failu pārvaldība;
- testēšana;
- realizē integrētas programmēšanas valodu versiju kontroles sistēmas;
- plug-in sistēma, kas ļauj pievienot papildus funkcionalitāti un atbalstu dažādām programmēšanas valodām un rīkiem.

Eclipse IDE ir plaši izmantota kā Java programmēšanas vide un tā ir pieejama daudzās operētājsistēmās, tostarp Windows, MacOS un Linux.

“Automatizēta dzīves apraksta izveide” pētniecība un izstrāde tiek veikta platformā Eclipse IDE. Veicot izpēti, kura platforma ir pati parocīgākā šim pētnieciskajam darbam, tika atrasta mājaslapa [12] ar informāciju par Eclipse IDE priekšrocībām. Autors piekrita vairumam izvirzīto secinājumu, kas tika izvirzīti par labu Eclipse IDE izvēlei.

1.5. Kopsavilkums

Teorētiskajā daļā ir apkopota informācija par programmēšanas valodu Java, tās priekšrocībām salīdzinot ar citām plaši zināmām, pazīstamām programmēšanas valodām. Tiek atspoguļota informācija par CV būtību un izmantotajiem resursiem pētījumā. Tiek iepazīstināts ar galveno rīku mērķa veiksmīgai sasniegšanai - Apache POI XWPF klase. Tiek pamatots Eclipse IDE lietojums.

2.Praktiskā daļa

Praktiskā daļā tiek aprakstīti programmas izveides soļi. Praktiskā daļa tiek sadalīti soļos un vizualizētas ar bildēm no programmas. Visus aprakstītos soļus ir iespējams saskatīt programmas kodā.

2.1 Bibliotēku izvēle

Pēc teorijas izstrādes, pirmais solis bija piemeklēt bibliotēkas. Darbā pakāpeniski tika piemeklētas programmai nepieciešamās bibliotēkas. Autors noskaidroja, ka trūkst funkciju, pamata bibliotēkā [10] - JRE System Library [Java SE-17] , kas nodrošinātu izvirzīto mērķi - programma izveido failu docx formātā. Tādēļ mērķa sasniegšanai tika izmantota papildus interneta bibliotēka Apache POI[6].

Pirmās programmā tika ievietotas JRE System Library [Java SE-17] pakotnes. Pamata bibliotēkas pakotnes neprasa īpašu uzmanību, to ievietošanā. Pēc pakotnes nosaukuma ir iespējams identificēt, no kuras bibliotēkas tā ņemta, kā piemēram, java bibliotēkām ir (java.), bet Apache POI bibliotēkām (org.apache.poi). (*skatīt 1. attēlā*).

```

3 import java.io.FileOutputStream;
4 import java.io.OutputStream;
5 import java.util.Scanner;
6
7 import org.apache.poi.xwpf.usermodel.ParagraphAlignment;
8 import org.apache.poi.xwpf.usermodel.XWPFDocument;
9 import org.apache.poi.xwpf.usermodel.XWPFPParagraph;
10 import org.apache.poi.xwpf.usermodel.XWPFRUN;
11 import org.apache.poi.xwpf.usermodel.XWPFTable;
12 import org.apache.poi.xwpf.usermodel.XWPFTableRow;

```

1.attēls. Programmas piemērs. Autors Jana Ozolniece

2.2 CV jautājumu izveide

Nākamais solis pēc kārtas - izveidot jautājumus pēc atrastā parauga jautājumiem. Programmā tika ievietoti jautājumi, kuri atklāj CV prasīto informāciju saprotamā veidā. Jautājumu izveidē, tika izmantotas komandas no pamata bibliotēkas.

Programma izveidota tā, lai lietotājs konsolē var ievadīt savas atbildes uz dotajiem jautājumiem, kuri minēti punktā (1.2). Lietotāja ievadīto datu saglabāšanai, tika izmantota skeneru klase no pamata bibliotēkas util.Scanner pakotnes (*skatīt 1. attēlā*). Gandrīz katram konsolē uzdotajam jautājumam tiek piedāvāts piemērs, lai veicinātu jautājuma saprašanu un pārdomāšanu.

2.3 Datu saglabāšana

Nākamais solis - saglabāt lietotāja ievadītos datus. Ievāktās atbildes uz (1.2) punkta jautājumiem, tika saglabātas virknē (String)(*skatīt 2. attēlā*). Izmantojot operatoru (if, else) , atbildes tika loģiski šķīrotas (*skatīt 3. attēlā*). Tika lietotas viendimensiju matricas un (for) cikls, lai saglabātu un koriģētu datus par valodas zināšanām(*skatīt 3. attēlā*). Lietotāja atbilžu datus bija iespējams saglabāt ar pamata bibliotēku(2.1).

Viendimensiju matricas tika izmantotas jautājumiem, kuriem atbildes varēja tikt sniegtas daudzas un dažādas, kā piemēram, jautājumi par darba pieredzi, kur tā iegūta,

jautājumi par valodas prasmēm, jautājumi par citām prasmēm, kuras lietotājs vēlētos uzrādīt CV.

```
String vards,atb,dzivo,epas,vakc,izgl,sasn,papiz,citaspr;
String Aa = null,v= null,p= null;
int m;
```

2.attēls.Programmas piemērs. Autors Jana Ozolniece

```
String[] valoda = new String[dati.nextInt ()];
dati.nextLine ();
for (int i = 0; i < valoda.length; i++)
{
    System.out.println ("ievadiet valodu (Piem., Latviešu)");
    valoda[i] = dati.nextLine ();
}
```

3.attēls.Programmas piemērs. Autors Jana Ozolniece

2.4 Word dokumenta izveide

Nākamajā solī tika veikta dokumenta izveide. Izmantojot teorijā savāktu informāciju par XWPF klasi, tika izvēlētas vairākas komandas, kā *XWPFDocument*, *XWPFParagraph* un *XWPFRun*, kuras secīgi izveidoja dokumentu, izveidoja jaunu rindkopu un definēja teksta apgabalu ar kopīgu īpašību kopu (*skatīt 4. attēlā*). Lai pareizi izmantotu šīs komandas, tika izpētīts paraugs [7].

Dokumenta saglabāšanai uz darba virsmas, bija nepieciešams lietot *OutputStream* un *FileOutputStream* klases, lai sekmīgi saglabātu dokumentu datorā. Lietotāja datorā word dokumenta nosaukums tika saglabāts, kā lietotāja ievadītais vārds un uzvārds, pievienojot (cv.dokuments) beigās. (*skatīt 4. attēlā*).

```
XWPFDocument dokument = new XWPFDocument();
try(OutputStream os = new FileOutputStream(vards+" cv.dokuments")) {
    XWPFParagraph paragraphs = dokument.createParagraph();
    XWPFRun runcv = paragraphs.createRun();
```

4.attēls.Programmas piemērs. Autors Jana Ozolniece

2.5 Datu ievietošana word dokumentā

Izmantojot komandas *add.break*, *setFontSize*, *setBold*, *setItalic* un *setText*, bija iespējams rediģēt tekstu izveidotajā word dokumentā (*skatīt 5. attēlā*). Komanda *setAlignment* ļāva manipulēt ar teksta izvietojumu uz lapas (*skatīt 5. attēlā*).

Lietojot *XWPFTable* un *XWPFTableRow* klasi [8,9], tika izveidota tabula. Lai ievietotu datus tabulā, tika izmantotas komandas, tādas kā *getCell*, *setText* un *addNewTableCell* [8,9], lai secīgi lasītu datus kolonā, ievietotu tekstu un izveidotu jaunu rindu (*skatīt 6. attēlā*).


```

paragrafs = dokuments.createParagraph();

paragrafs.setAlignment(ParagraphAlignment.LEFT);
XWPFRun run = paragrafs.createRun();
run.addBreak();          run.addBreak();
run.setFontSize( 14);
run.setBold(true);
run.setItalic(true);
run.setText("Vārds Uzvārds: "+vards);
run.addBreak();

```

5.attēls. Programmas piemērs. Autors Jana Ozolniece

```

XWPFTable tabula = dokuments.createTable();
XWPFTableRow rinda = tabula.getRow(0);

run.setText("Valodu prasmes");
run.addBreak();

tabula.setWidth("100%");

rinda.getCell(0).setText("valoda");
rinda.getCell(0).getParagraphArray(0).setSpacingAfter(0);
rinda.addNewTableCell().setText("runāšana prasmes");
rinda.getCell(0).getParagraphArray(0).setSpacingAfter(0);
rinda.addNewTableCell().setText("saprāšanas prasmes");
rinda.addNewTableCell().setText("rakstīšanas prasmes ");

```

6.attēls. Programmas piemērs. Autors Jana Ozolniece

3.Rezultātu analīze

Lai secinātu, ka programma veic savu funkciju, ka pētījuma mērķis tiek sasniegts, programmu testēja dažādi lietotāji. Lietotāju atsauksmes tika apkopotas. No iegūtajām atsauksmēm tika izveidota rezultātu analīze. Pielikumos ir programmas izveidotie testētāju CV .

1. testējamā atsauksme

Pirmajam testējamajam tika uzdoti jautājumi par programmas kvalitāti:

- vai programma bija ērti lietojama,
- vai jautājumi bija saprotami,
- vai šādi ir vieglāk sastādīt CV, salīdzināt ar citām vietnēm.

Atbildes uz jautājumiem bija apstiprinošas, par labu programmai. Diemžēl testētā CV nav pašlaik pieejams apskatīšanai.

2. testējamā atsauksme

Otrais testējamais programmas izpildes laikā piedzīvoja problēmas ar lasītprasmi. Redzot, šīs problēmas, programma tika pārveidota saprotamākā veidā. Pēc programmas izpildes un CV dokumenta apskates, testējamajam tika uzdoti jautājumi:

- vai programma ir ērti lietojama,
- vai jūs programmu lietu nākotnē,
- ko jūs domājat par šo programmu.

Testējamā sniegtās atbildes bija īsas un kodolīgas. Pēc ierakstā fiksētā, tika noformulēta atsauksme. Atsauksme ir pozitīva, testējamais liecināja, ka rīks ir noderīgs un ērti lietojams. *(skatīt pielikuma 2. punktu).*

3. testējamā atsauksme

Trešajam testējamajam programmas dotie jautājumi nesagādāja problēmas atbildēt. Programmas dotie jautājumi radīja pārdomas par profesionālo pieredzi, par nākotnes profesijas izvēli.

Testējamais, pēc pabeigta CV dokumenta redzēšanas, bija patīkami savīļņots. Testējamais izteica komentārus, kā "Lūk, cik šis ir brīnišķīgi", "es esmu pilnīgā šokā", "tagad saprotu, kāda bija jautājumu nozīme", "ļoti labs rīks, man patīk , jo, kad es rakstīju CV, es biju ļoti apjucis par to, kas jāraksta", "šī programma atvieglo darbu tiem, kuri nesen beiguši skolu un nekad nav rakstījuši CV". Testējamajam jautājumi netika uzdoti. *(skatīt pielikuma 3. punktu).*

Secinājumi

1. Programmu, kas izveido, modificē word dokumentu, nav iespējams izveidot ar pamata bibliotēku - JRE System Library [Java SE-17].
2. No interneta resursiem ievietojot bibliotēku, ir vērts pameklēt informāciju par veidu, kā to ievietot programmā.
3. Programmu visvieglāk veidot ir, sadalot to daļās - jautājumi, darbības, atklūdošana.
4. Izmantojot Apache POI, ir svarīgi izpētīt norādīto dokumentāciju, ar kuru tiks strādāts, lai veiksmīgi izpildītu norādītās komandas.
5. Programmu, kas automātiski izveido CV dokumentu, ir iespējams izveidot pat nepieredzējušiem programmētājiem.

Literatūras saraksts

- [1] https://www.java.com/en/download/help/whatis_java.html(12.01.2023)
- [2] <https://www.cv.lv/lv/>(07.01.2023)
- [3] <https://i.pinimg.com/736x/eb/e0/8c/ebe08cb556f7648e9ba3a4e4f768da90.jpg>
(09.012023)
- [4] <https://poi.apache.org/components/> (09.012023)
- [5] <https://poi.apache.org/components/document/index.html> (12.01.2023)
- [6] <https://poi.apache.org/download.html#POI-5.2.3> (12.01.2023)
- [7].<https://www.javatpoint.com/apache-poi-creating-ms-word> (09.012023)
- [8]<https://poi.apache.org/apidocs/dev/org/apache/poi/xwpf/usermodel/XWPFTable.html>
(09.012023)
- [9] (09.012023)
<https://poi.apache.org/apidocs/dev/org/apache/poi/xwpf/usermodel/XWPFTableRow.html>
- [10]
<https://textbooks.cs.ksu.edu/cc310/12-performance/10-java-libraries/1-1-java-standard-library/> (12.01.2023)
- [11]<https://www.educba.com/what-is-eclipse-ide/> (12.01.2023)
- [12]<https://www.educba.com/what-is-eclipse-ide/> (12.01.2023)

Pielikumi

1. Pielikums

ZPD "Automatizēta dzīves apraksta izveide.

Saite uz Google drive, kur parādīts kā palaist programmu.

<https://drive.google.com/drive/folders/1oNFuFEFbGyYPje71UG6-YjdK1Wlomfel?usp=sharing>

CV

Vārds Uzvārds: Testējamais 2

Tālrunis: 1234567

Adrese: Purvciema iela 10, Pilsētiņa

E-Pasts: pēterītis@zpd.com

CV Mērķis : Diriģents

Augstākā izglītība iegūta:

Ventspils mūzikas akadēmija

Darba Pieredze: Diriģents, S.Guntars, 2015-2015

Profesionālie sasniegumi: Labākais gada darbinieks

Papildus izglītība:

nav

Citas prasmes:

Diriģēšana

Klavierspēle

Valodu prasmes:

valoda	runāšanas prasmes	saprašanas prasmes	rakstīšanas prasmes
Latviešu	labi	izcili	Teicami
Angļu	optimāli	optimāli	Optimāli

CV

Vārds Uzvārds: Testējamais 3

Tālrunis: 1234567

Adrese: Lilliju iela 420, Pilsēta

E-Pasts: testejamais3@zpd.lv

CV Mērķis : Medicīna

Pamatizglītība iegūta:

Talsu Valsts ģimnāzija

Darba Pieredze: Aktieris, K.Mikus, 2021 - pašlaik

Profesionālie sasniegumi: Galvenā loma izrādē

Papildus izglītība:

Mūzikas skola

Citas prasmes:

Klavierspēle

Valodu prasmes:

Valoda	runāšanas prasmes	saprašanas prasmes	rakstīšanas prasmes
Latviešu	teicami	teicami	teicami
Krievu	teicami	teicami	teicami
Angļu	labi	labi	labi