

Sprawozdanie

Projekt
"Kwiaciarnia"

Wykonane przez: Jarosław Jędruszcak

1 . Wprowadzenie

Tematem projektu jest wykorzystanie pozyskanej wiedzy w tworzeniu programu który na podstawie podanych danych jest w stanie wyświetlić sugestie na podstawie podobieństw między danymi podanymi wcześniej. Tematyka dla mojego projektu to "Kwiaciarnia", program który na podstawie wpisanej frazy wyświetla dostępne kwiaty, a po wybraniu szukanego kwiatu wyświetla w formie reklamy inne kwiaty które zostały w podobny sposób wyszukane.

2 . Cel Projektu

Celem projektu jest uzyskanie indywidualnych reklam dla każdej różnej wpisanej frazy i wybranego kwiatu na podstawie tworzącej się wewnętrznie macierzy miar odległości Euclidesa gdzie termami są słowa wpisane przy przeszukiwaniu inwentarza kwiaciarni, a porównywane są dostępne produkty w kwiaciarni które posiadają tagi służące do odnajdywania produktów podczas przeszukiwania inwentarza oraz na podstawie poprzednich przeszukiwań zlicza występowanie tagów we wpisywanych frazach które prowadzą do wybrania tego przedmiotu.

3 . Działanie podprogramu udzielającego sugestie

Sugestie wyświetlane są pod postacią nazw produktów które według miar odległości Euclidesa były najbardziej podobne do wbranego produktu ale tylko jeżeli tagi dla tego polecanego produktu zostały użyte przynajmniej 3 razy. Wynikiem są maksymalnie 3 pozycje z nazwami polecanych produktów

4 . Działanie programu

- Program bazuje na dwóch plikach tekstowych:

kwiaty.txt – plik przechowuje dane o dostępnych produktach kwiaciarni.

Nazwa kwiatu	Tagi
Irys.	IRYS, FIOLET, FIOLETOWY, KWIAT

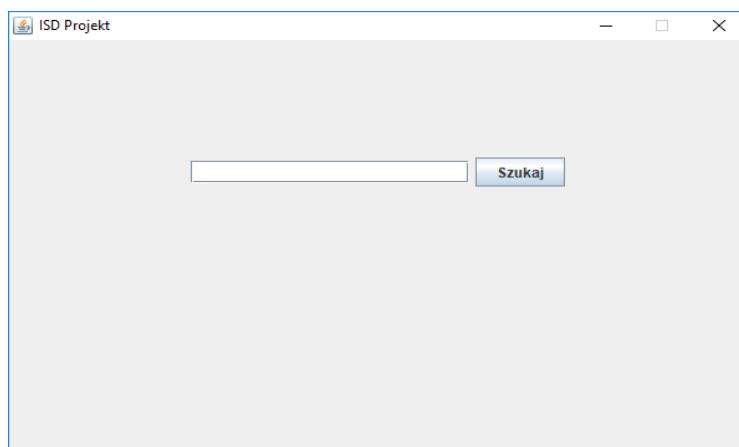
Nazwa kwiatu jest otrzymywana poprzez oddzielenie całego słowa używając kropki "." jako punkt oddzielający nazwę od tagów. Następnie tagi rozdzielane są przecinkiem. Każda linijka pliku jest zapisywana w obiekcie klasy Produkt który umieszczony jest w liście obiektów tej klasy.

wyszukaneDane.txt – plik przechowuje dane o poprzednich frazach użytych w wyszukiwarce

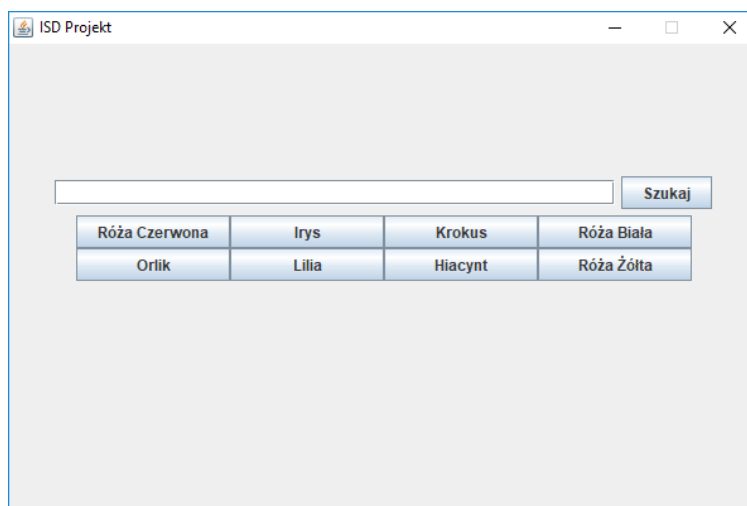
Tagi	Nazwa
ZOLTY, KWIAT.	Krokus

Tagi zostały uzyskane na podstawie frazy użytej podczas przeszukiwania, a nazwa to produkt który został wybrany po uzyskaniu wyniku przeszukiwania inwentarza.

- Okno wyszukiwarki



Aby wyszukać interesujący nas kwiat nasza wpisywana fraza lub jej fragment musi się pokrywać z jednym z tagów danego kwiatu. Na przykład po wpisaniu słowa "kwiat" wynikiem będzie wyświetlenie się ośmiu przycisków po jednym dla każdego z ośmiu kwiatów. Dzieje się tak ponieważ 8 z 10 produktów ma w liście tagów słowo "kwiat".

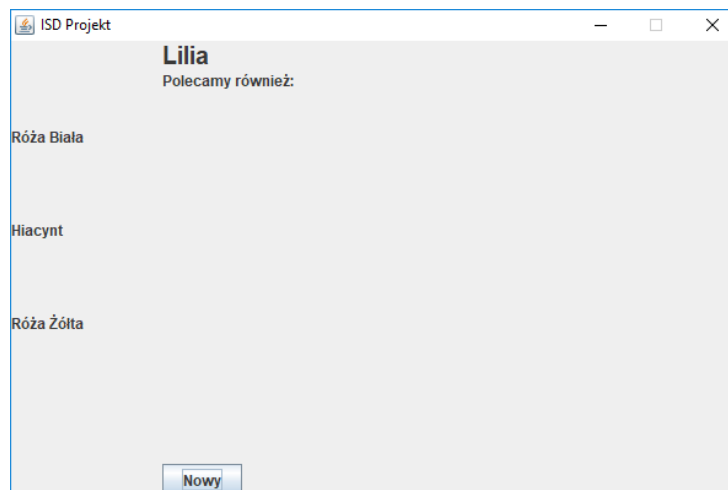


```
Irys.IRYS,FIOLET,FIOLETOWY,KWIAT
Róża Czerwona.ROZA,CZERWONA,CZERWONY,KWIAT
Róża Biała.ROZA,BIAŁA,BIAŁY,KWIAT
Róża Żółta.ROZA,ZOLTA,ZOLTY,KWIAT
Lilia.LILIA,ZOLTA,ZOLTY,KWIAT
Hiacynt.HIACYNT,FIOLET,FIOLETOWY,ROZOWY,ROZ,KWIAT
Krokus.KROKUS,ZOLTY,ZOLTA,FIOLET,FIOLETOWY,KWIAT
Orlik.ORLIK,ROZ,ROZOWY,KWIAT
Bukiet rubinowy.BUKIET,RUBINOWY,CZERWONY,DUZY
Zestaw Sto lat.ZESTAW,STO,LAT,BUKIET,POMARANCZOWY,MALY
```

Dodatkowo jeżeli zmieniona zostanie fraza na "kwiat bukiet" spowoduje to wypisanie wszystkich przedmiotów.

- Okno produktu

Po wybraniu jednego z przedmiotów otwierana jest strona na której widnieje nazwa wybranego kwiatu oraz lista trzech polecanych innych kwiatów.



Widoczne reklamowane produkty zostały wybrane na podstawie powstałej macierzy reprezentującej miary odległości euclidesa gdzie termem jest "kwiat" danymi są wszystkie produkty a celem porównań jest kwiat o nazwie "Lilia".

	Róża czerwona	Irys	Krokus	Róża biała	Orlik	Zestaw Sto Lat	Lilia	Hiacynt	Bukiet rubinow	Róża Żółta
kwiat	16	64	16	9	16	81	0	9	81	9

Powyższy obrazek nie pokazuje miary odległości euclidesa ale pokazuje kwadraty różnic występowania termu "kwiat" między kwiatem "Lilia", a pozostałymi.

- Zapis wyszukiwanych danych

Zapis odbywa się podczas klikania na przycisk reprezentujący szukany kwiat. Zapisywana jest zależność między znalezionym kwiatem, a frazą użytą do wyszukania tego kwiatu i jest ona zapisana w następujący sposób:

1. Fraza użyta do szukania : "kwiat niebieski"
2. Wybrano kwiat "Róża czerwona"
3. W pliku powstaje wpis: "KWIAT,NIEBIESKI.Róża czerwona"

- Odczyt z pliku

Odczyt odbywa się po uruchomieniu programu oraz po kliknięciu przycisku "Nowy" w Oknie Produktu. Odczyt zaczyna się od czytania pliku wyszukaneDane.txt (dla tego indeksy niezgadza się z kolejnością widniejącą w pliku kwiaty.txt) na podstawie każdego wpisu tworzony jest obiekt klasy Produkt który zawiera wszystkie informacje na temat danego kwiatu, jeżeli już taki obiekt istnieje to dodaje występujące w wpisie tagi do odpowiedniego obiektu. W przypadku gdy wpis w pliku wyszukaneDane.txt posiada tag który jest nie rozpoznawalny według pliku kwiaty.txt, tag nie jest zapisywany ponieważ gdyby tagi były dynamicznie tworzone dla produktów można by było po praru wpisach np.: wywołać produkt "Bukiet rubinowy" hasłem "kwiat" co było by problematyczne i prowadziło by do błędów.

- Wybieranie reklamowanych kwiatów

Do wybierania reklam użyte zostały dwa array'e trzy elementowe. Jeden zawiera wartość odległości euclidesowej i zostaje wywołana z wypełnionymi komórkami liczba o 1 większą od maksymalnej wartości odległości euclidesa. Drugi zawiera indeksy wskazujące na indeks w liście obiektów klas Produkty i zostaje wywołana z wypełnionymi komórkami liczba -1. Przy wybieraniu reklam sprawdzane jest czy suma występowania termów dla danego produktu jest większa od 4, jeżeli jest to kolejno zostaje porównywana ze wszystkimi komórkami array'u zawierającego wartości odległości Euclidesowej, jeżeli wartość komórki jest mniejsza to ta wartość jest przekazywana dalej do kolejnych komórek array'a i zostaje zastąpiona większą i zapisywany jest do komórki array'u z indeksami indeks produktu.

Po zakończeniu array zawierający indeksy jest sprawdzany. Jeżeli komórka jest różna od "-1" to zostaje zamieszczona reklama do produktu o indeksie przechowywanym przez komórkę.

5 . Kod programu

- Metoda `wyswietlKwiaty` wyświetla przyciski, tworzy macierz TFM ...

```
public static void wyswietlKwiaty()
{
    wyszukiwanaFraza = wyszukiwanaFraza.toUpperCase().replace('ż','Z');
    wyszukiwanaFraza = wyszukiwanaFraza.toUpperCase().replace('ó','O');
    wyszukiwanaFraza = wyszukiwanaFraza.toUpperCase().replace('ę','E');
    wyszukiwanaFraza = wyszukiwanaFraza.toUpperCase().replace('ł','L');
    wyszukiwanaFraza = wyszukiwanaFraza.toUpperCase().replace('ą','A'); // Zamieniam Polskie znaki poniewaz dane zapisuje bez nich

    String[] PodzielonaWF = wyszukiwanaFraza.split(Pattern.quote(" ")); // Dzielę słowo podane przy wyszukiwaniu na termsy

    int x = 0;
    int y = 0;
    int[][] tablicaTFM = new int[PodzielonaWF.length][ListaWystapienProduktow.size()]; // Tworzy array na podstawie ilosci termow i produktow

    for (int x1 = 0; x1 < tablicaTFM.length; x1++) {
        for (int y1 = 0; y1 < tablicaTFM[x1].length; y1++) { // Inicjuje Array
            tablicaTFM[x1][y1] = 0;
        }
    }

    String[] NaglowkiTFM = new String[ListaWystapienProduktow.size()];
    String[] ListaTagow;
    int[] ListaWystapien;

    for (Produkt produkt : ListaWystapienProduktow)
    {
        x = 0;
        ListaTagow = produkt.getListaTagow();
        ListaWystapien = produkt.getListaWystapienTagow();

        for (String slowo : PodzielonaWF) {
            int index = 0;
            for (String nazwa : ListaTagow) //dla kazdego elementu z listy obiektow produktu
            {
                if (slowo.equals(nazwa))
                {
                    tablicaTFM[x][y] = ListaWystapien[index]; //dodaje dane do macierzy TFM
                }
                index ++;
            }
            x++;
        }
        NaglowkiTFM[y] = produkt.getNazwa(); // zapisuje naglowek osobno
        y++;
    }

    int[][] TablicaEuq = new int[PodzielonaWF.length][ListaWystapienProduktow.size()];
    int[] OdleglosciEuq = new int[ListaWystapienProduktow.size()];
}
```

... oraz tworzy przyciski dla kazdego znalezionej kwiatu.

```
for (Produkt produkt : ListaWystapienProduktow) // dla kazdego produktu z listy obiektow
{
    wychodzexD: // tag dla break'a
    for (String tag: produkt.getListaTagow()) // dla kazdego tagu z listy tagow
    {
        for(String Fraza : PodzielonaWF) // oraz dla kazdego slowa podanego podczas wyszukiwania
        {
            if (tag.equals(Fraza)) { // jezeli tag jest rowny frazie
                JButton przycisk = new JButton(String.valueOf(produkt.getNazwa())); // tworzy przycisk
                przycisk.addActionListener(new ActionListener() { // oraz dodaje ActionListenera
                    //
                    public void actionPerformed(ActionEvent e) {}
                });
                PanelGridow.add(przycisk); // dodaje przycisk do panelu gridow
                break wychodzexD; // aby uniknac duplikatow wylamuje sie z 2 petli na raz ( do tagu )
            }
        }
    }
}
```

- ActionListener zawarty w metodzie wyswietlKwiaty jest tworzony dla każdego z przycisków. Tworzy macierze powiązane z miarami odległości Euclidesa, zapisuje wcześniej zczytaną zależność między frazą użytą do wyszukania kwiatu a samym kwiatem ...

```
public void actionPerformed(ActionEvent e) {
    nazwaPrzedmiotuJakoNaglowek.setText(produkt.getNazwa());

    Writer output;          // Dodaje wpis do piku wyszukaneDane.txt
    try {
        String tagowanie = "";
        for (String slowo : PodzielonaWF)
        {
            tagowanie = tagowanie + slowo + ",";
        }
        tagowanie = tagowanie.substring(0, tagowanie.length() -1);
        output = new BufferedWriter(new FileWriter(plikWyszukiwan, true));
        output.append(tagowanie + "." + produkt.getNazwa());
        output.append(System.getProperty("line.separator"));
        output.close();
    } catch (IOException e1) {}

    int indexSzukanegoTaga = Arrays.asList(NaglowkiTFM).indexOf(produkt.getNazwa()); // zwraca indeks produktu

    for (int x = 0; x < tablicaTFM.length; x++) {          // tworzy macierz przygotowująca dane dla odleglosci Euclidesowych
        for (int y = 0; y < tablicaTFM[x].length; y++) {    // ten array jest wyswietlany razem z wynikiem(reklamami)
            TablicaEuq[x][y] = (int) Math.pow(tablicaTFM[x][indexSzukanegoTaga] - tablicaTFM[x][y], 2);
        }
    }

    double[] WlasciweWartosciEuq = new double[OdleglosciEuq.length];
    int index = 0;

    for (int x = 0; x < tablicaTFM.length; x++) {          // sumuje odleglosci
        for (int y = 0; y < tablicaTFM[x].length; y++) {
            OdleglosciEuq[y] = OdleglosciEuq[y] + TablicaEuq[x][y];
        }
    }

    double max = 0;

    for (double wartoscEuq : WlasciweWartosciEuq) {        // pierwiastkuje zsumowane odleglosci tworzac miary odleglosci Euclidesa
        wartoscEuq = Math.sqrt(new Double(OdleglosciEuq[index]));
        if (max < wartoscEuq) max = wartoscEuq;
        WlasciweWartosciEuq[index] = wartoscEuq;
        index ++;
    }
    max ++;
}
```

... wybierane są reklamy ...

```
index = 0;
double ii;

double reklamy[] = {max,max,max};

int[] indexyReklam = {-1,-1,-1};

// W tym bloku maja zostac wybrane produkty do zareklamowania na podstawie ich odleglosci od wskazanego produktu 3 najblizesze zostaja
stop:
for (double i : WlasciweWartosciEuq)
{
    ii = i;
    if (OdleglosciEuq[index] > 4)
    {
        if (ii < reklamy[0])
        {
            reklamy[2] = reklamy[1];
            reklamy[1] = reklamy[0];
            reklamy[0] = ii;
            indexyReklam[2] = indexyReklam[1];
            indexyReklam[1] = indexyReklam[0];
            indexyReklam[0] = index;
            //break stop;
            index ++;
            continue stop;
        }

        if (ii < reklamy[1])
        {
            reklamy[2] = reklamy[1];
            reklamy[1] = ii;
            indexyReklam[2] = indexyReklam[1];
            indexyReklam[1] = index;
            index ++;
            //break stop;
            continue stop;
        }

        if (ii < reklamy[2])
        {
            reklamy[2] = ii;
            indexyReklam[2] = index;
            index ++;
            //break stop;
            continue stop;
        }
    }
    index ++;
}
```


... dodawane są reklamy do interfejsu oraz zostaje wyświetlone pomocnicze okno do sprawdzania poprawnego działania programu.

```
GridPanel2.removeAll(); //Resetuje panel przed dodaniem do niego nowych elementow
GridPanel2.revalidate();
GridPanel2.repaint();

boolean reklamowac = false;

for (int i : indexyReklam) // Kazdy element array'a indexyReklam jest sprawdzany
{
    // Jezeli iszy element jest rowny -1 nie zostana podane zadne reklamy
    if (i != -1)
    {
        JLabel ReklamowanyProdukt = new JLabel(NaglowkiTFM[i]);
        GridPanel2.add(ReklamowanyProdukt);
        reklamowac = true; // Jezeli istnieje co najmniej 1 reklama dodaje Label z "Polecamy rowniez"
    }
}

if (reklamowac)
    PolecaneLabel.setVisible(true);
else
    PolecaneLabel.setVisible(false);

JFrame rezultat = new JFrame("Euqlides"); //Dodatkowo dodalem nowe okienko z wziesniej wspomnianym arrayem
//jako tabelka przedstawiajaca wartosci odleglosci
String[][] MEuqKonw = new String[PodzielonaWF.length][ListawystapienProduktow.size()];

for (int x = 0; x < MEuqKonw.length; x++) {
    for (int y = 0; y < MEuqKonw[x].length; y++) {
        MEuqKonw[x][y] = Integer.toString(TablicaEuq[x][y]);
    }
}

JTable TEuq = new JTable(MEuqKonw, NaglowkiTFM);

rezultat.setSize(new Dimension(500, 150));
rezultat.add(TEuq);

rezultat.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
rezultat.setVisible(true);
|
przelacznik.show(_1stLayer, "Po_wybraniu_kwiata");
```

6 . Podsumowanie

Program poprawnie wyświetla reklamy na podstawie fraz wpisanych podczas wyszukiwania i wybranego wyniku wyszukiwania. Dzieje się to przez wykorzystanie miar odległości Euqlidesa które to pozwoliły na określenie odległości podobieństw między produktami. Dzięki czemu uzyskałem odpowiednie dane które mogłem wykorzystać tak aby program działał tak jak chciałem.