Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ОТЧЕТ

к лабораторной работе по дисциплине:

|  |
| --- |
| **Методы и средства проектирования информационных систем и технологий** |
| Поддержка принятия решений в задачах группового (коллективного) выбора |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСМб-19-1 |  |  |  | Михейко Ю.А. |
|  |  | подпись |  | Фамилия И.О |
| Проверил: |  |  |  |  | Юрин А.Ю. |
|  | должность |  | подпись |  | Фамилия И.О |

Иркутск 2022 г.

Содержание

[1 Постановка задачи 3](#_Toc118802954)

[2 Описание тестовых примеров 4](#_Toc118802955)

[3 Описание реализуемого метода группового выбора 6](#_Toc118802956)

[3.1 Неформальное описание 6](#_Toc118802957)

[3.2 Блок-схема 6](#_Toc118802958)

[4 Результаты решения тестовых задач 7](#_Toc118802959)

[5 Фрагмент сгенерированной документации 8](#_Toc118802960)

[6 Листинг кода 9](#_Toc118802961)

1 Постановка задачи

Цель работы: выработка и закрепление навыков по использованию систем и модулей поддержки принятия решений в задачах группового (коллективного) выбора.

Для выбранного варианта задания необходимо:

1. Разработать 6 тестовых примеров для задач группового (коллективного) выбора, при этом 2 примера должны содержать парадоксы голосования.

2. Решить разработанные примеры с использованием СППР КВ.

3. Произвести разработку собственного программного модуля, реализующего метод (согласно варианту задания).

4. Решить примеры с использованием разработанного модуля в составе СППР КВ.

5. Сгенерировать комплект документации (отчет по решенным задачам).

**Вариант**

|  |  |  |
| --- | --- | --- |
| N | Метод | Краткое описание |
| 8. | процедура Доджсона (Dodgson) | все альтернативы упорядочиваются по числу голосов, которых им не хватает для того, чтобы превосходить все остальные варианты по простому большинству (более половины) голосов, упорядочение строится по возрастанию значений |

2 Описание тестовых примеров

Для выполнения данной рабораторной работы было составлено шесть тестовых примеров. Два примера, согласно заданию, содержали парадоксы.

**Пример 1**

Выбор сериала.

**Описание:**

Выбор сериала для просмотра.

**Альтернативы:**

a=Черное зеркало

b=Асока

c=Время приключений: Фионна и Кейк

**Ранжировки:**

a-b-c=8

b-a-c=10

с-a-b=13

**Пример 2**

Выбор города для отдыха.

**Описание:**

Выбор города для отдыха на основе личных предпочтений.

**Альтернативы:**

a=Санкт-Петербург

b=Лондон

c=Чикаго

**Ранжировки:**

c-a-b=15

a-c-b=10

c-b-a=7

**Пример3**

Выбор музыки

**Описание:**

Выбор музыкального жанра для прослушивания.

**Альтернативы:**

a=Русский рок

b=Блюз

c=Кантри

**Ранжировки:**

c-b-a=17

c-a-b=5

b-c-a=12

**Пример 4**

Выбор игры

**Описание:**

Выбор игры для развлечения.

**Альтернативы:**

a=Sims 3

b=Warcraft 3

c= Plants vs. Zombies

**Ранжировки:**

a-b-c=8

b-a-c=15

c-a-b=9

**Пример 5 (Парадокс)**

Выбор такси.

**Описание:**

Выбор службы такси.

**Альтернативы:**

a=Яндекс

b=Maxim

c=Пойти пешком

**Ранжировки:**

a-с-b=15

b-c-a=12

c-b-a=20

**Пример 6 (Парадокс)**

Выбор иностранного языка

**Описание:**

Выбор иностранного языка для изучения.

**Альтернативы:**

a=Английский

b=Английский

c=Немецкий

**Ранжировки:**

a-с-b=23

b-c-a=19

c-b-a=19

3 Описание реализуемого метода группового выбора

3.1 Неформальное описание

В методе Доджсона каждый избиратель представляет упорядоченный список всех кандидатов в соответствии со своими предпочтениями (от лучших к худшим). Победителем считается кандидат, для которого нам необходимо выполнить минимальное количество парных обменов в каждом бюллетене (добавленное ко всем кандидатам), прежде чем они станут победителями по Кондорсе . В частности, если уже есть победитель Кондорсе , они побеждают на выборах.

3.2 Блок-схема

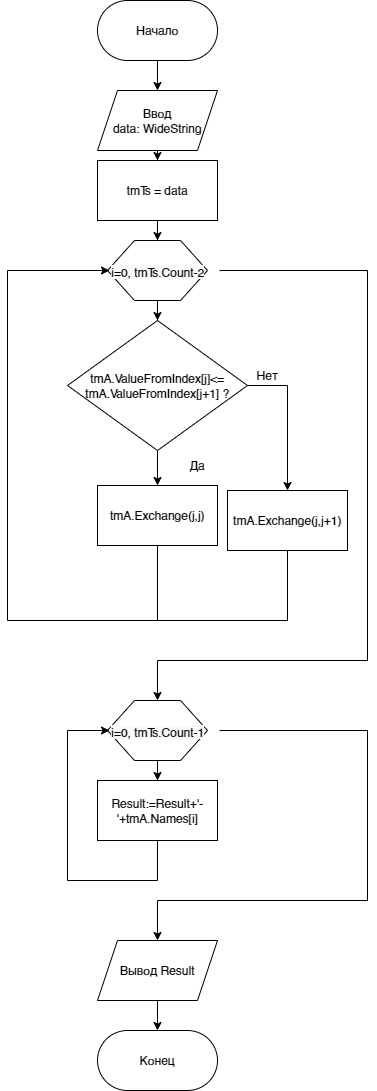


Рисунок 1 – Блок-схема метода

4 Результаты решения тестовых задач

В ходе решения тестовых примеров методом Доджсона с помощью программного средства СППР КВ и соответствующей готовой библиотеки были получены следующие результаты.

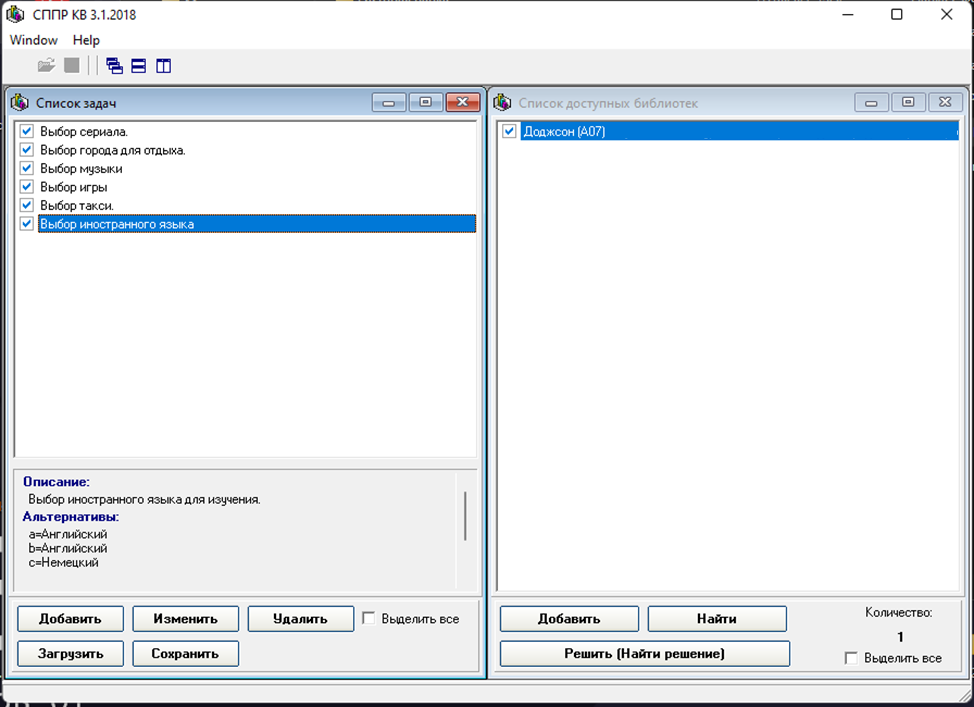


Рисунок 2 – Загрузка тестовых решений и метода Доджсона

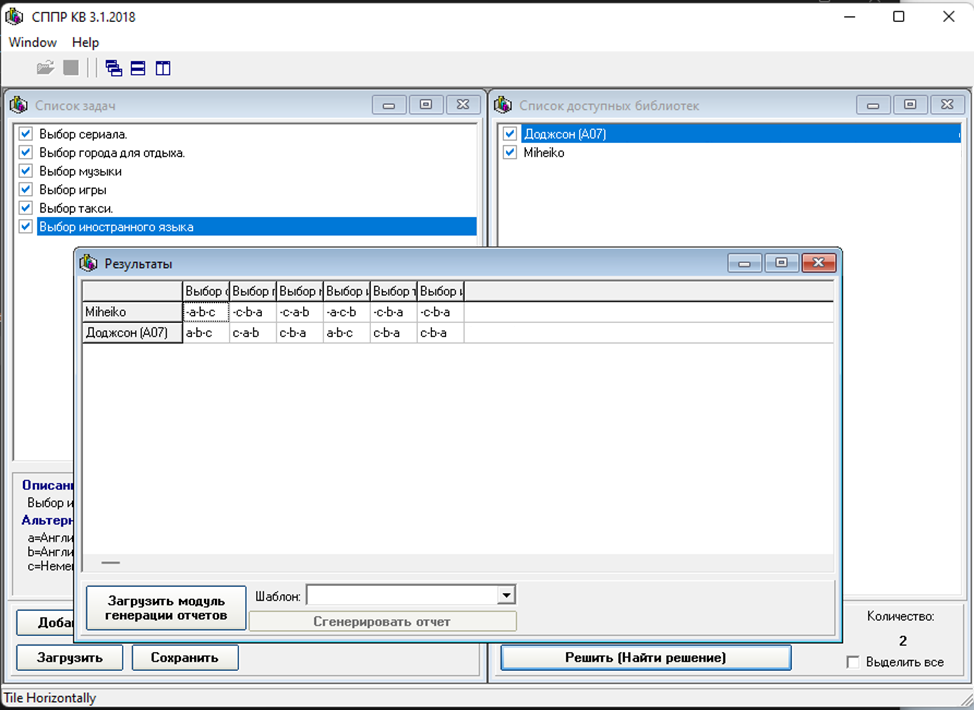


Рисунок 3 – Результат работы библиотек

В результате решения было выявлено, что разработанная библиотека выдает аналогичный результат в четырех из шести представленных задачах. Погрешности были выявлены в задачах-парадоксах.

5 Фрагмент сгенерированной документации

**Отчет № 08.11.2022**

**по задачам и методам**

Таблица 1 – Отчет о работе системы

|  |  |  |
| --- | --- | --- |
| Задача/ Метод | Miheiko | Доджсон (A07) |
| Выбор сериала. | -a-b-c | a-b-c |
| Выбор города для отдыха. | -c-b-a | c-a-b |
| Выбор музыки | -c-a-b | c-b-a |
| Выбор игры | -a-c-b | a-b-c |
| Выбор такси. | -c-b-a | c-b-a |
| Выбор иностранного языка | -c-b-a | c-b-a |

Количество задач: 6

Количество методов: 2

6 Листинг кода

library MyDll1;

{ Important note about DLL memory management: ShareMem must be the

first unit in your library's USES clause AND your project's (select

Project-View Source) USES clause if your DLL exports any procedures or

functions that pass strings as parameters or function results. This

applies to all strings passed to and from your DLL--even those that

are nested in records and classes. ShareMem is the interface unit to

the BORLNDMM.DLL shared memory manager, which must be deployed along

with your DLL. To avoid using BORLNDMM.DLL, pass string information

using PChar or ShortString parameters. }

uses

SysUtils,

Classes;

{$R \*.res}

function Execute (const data: WideString): WideString ; stdcall ;

var

i,j : Integer;

tmTs,tmTs1 : TStringList;

tmA : TStringList;

s,s1,s2 : ShortString;

begin

//

tmTs:=TStringList.Create;

tmTs1:=TStringList.Create;

tmA:=TStringList.Create; //list a

tmTs.Text:=data;

for i:=0 to tmTs.Count-1 do

begin

tmTs1.Delimiter:='-';

tmTs1.DelimitedText:=tmTs.Names[i];

s2:=tmTs.ValueFromIndex[i];

for j:=0 to tmTs1.Count-1 do

begin

s:=tmTs1.Strings[j];

if tmA.IndexOfName(s)=-1 then

tmA.Add(s+'=0');

if j=tmTs1.Count-1 then

begin

s1:=tmA.Values[s];

tmA.Values[s]:=IntToStr(StrToInt(s1)+StrToInt(s2));

end;

end;

end;

For i:=0 to tmA.Count-2 do

For j:=0 to tmA.Count-2 do

if tmA.ValueFromIndex[j]<= tmA.ValueFromIndex[j+1] then

tmA.Exchange(j,j)

else

tmA.Exchange(j,j+1);

Result:='';

For i:=0 to tmA.Count-1 do

Result:=Result+'-'+tmA.Names[i];

//Result:='0=0';

end;

function About: WideString; stdcall;

begin

Result := PChar('Miheiko');

end;

exports

About,

Execute;

begin

end.