Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Кафедра вычислительной математики и механики

**Лабораторная работа № 4**

**по дисциплине: «Интеллектуальные ИСИТ»**

Выполнил

студент группы ИСТ-19-2б

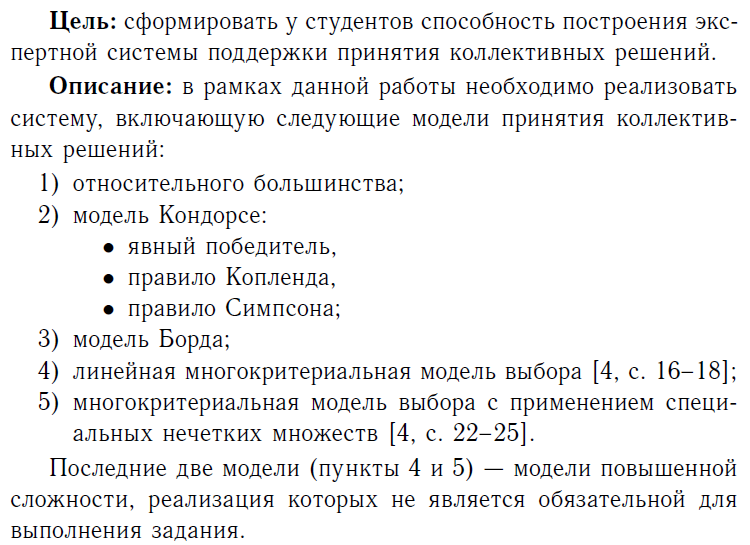
Семёнова А.С.

Проверил

ассистент кафедры ВММБ

Нетбай Г.В.

Пермь, 2022



**Описание**

1. Опрашиваются и заполняются кандидаты и избиратели;
2. Выбор модели (модель Борда /модель относительного большинства);
3. Проводится голосование;
4. Подсчёт голосов;
5. Выбор победителя.

Замечание: если сумма голосов будет равна, то выбрать победителя не получится. Голосование будет проводиться повторно.

**Модель Борда**

Всем избирателям предлагается проголосовать за кандидатов, выставив им места с 1 по p. За первое место присваивается p-1 балл. За последнее место присваивается 0 баллов. Затем голоса для каждого из кандидатов суммируются и выбирается кандидат с наибольшим количеством баллов. В случае равенства голосов-голосование проводится заново.

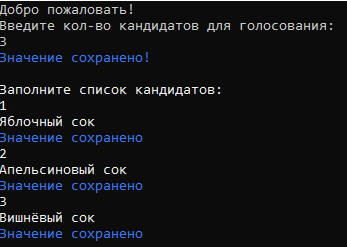
**Модель относительного большинства**

Всем избирателям предлагается выбрать одного из кандидатов. Затем подсчитывается количество голосов за каждого кандидата. Выбирается кандидат с наибольшим количеством баллов. В случае равенства голосов-голосование проводится заново.

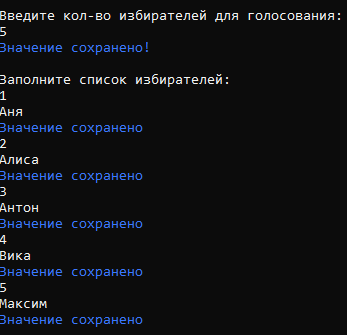
**Тестирование программы**

***Тестирование №1(Модель Борда)***

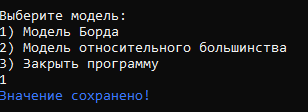
***Вводим количество кандидатов и заполняем список:***



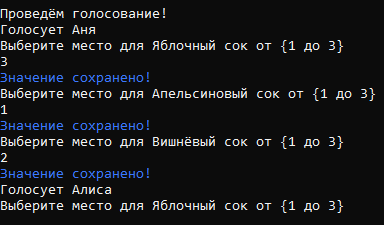
***Вводим количество избирателей и заполняем список:***



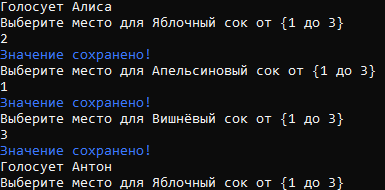
***Выбираем модель ПР:***



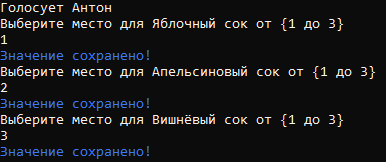
***Голосует избиратель №1(Аня):***



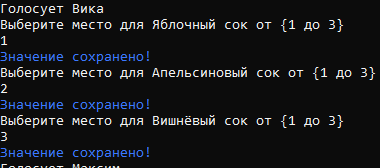
***Голосует избиратель №2(Алиса):***



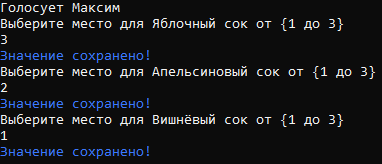
***Голосует избиратель №3(Антон):***



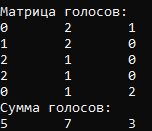
***Голосует избиратель №4(Вика):***



***Голосует избиратель №5(Максим):***



***Матрица и подсчёт голосов:***

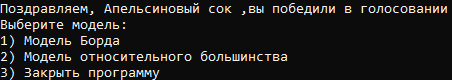


Яблочный сок-5

Апельсиновый сок-7

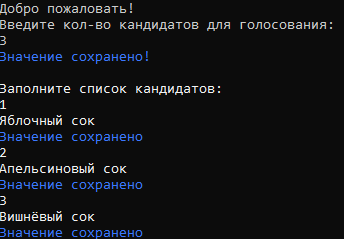
Вишнёвый сок-3

***Выбор победителя:***

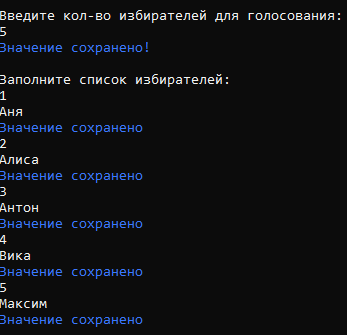


***Тестирование №2(Модель относительного большинства)***

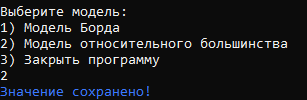
***Вводим количество кандидатов и заполняем список:***

******

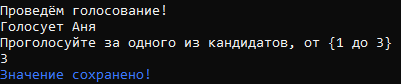
***Вводим количество избирателей и заполняем список:***



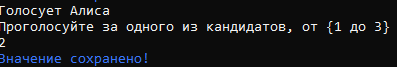
***Выбираем модель ПР:***



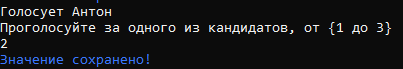
***Голосует избиратель №1(Аня):***

******

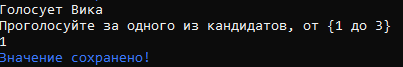
***Голосует избиратель №2(Алиса):***

******

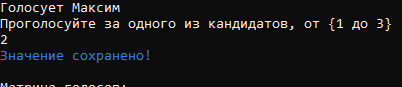
***Голосует избиратель №3(Антон):***

******

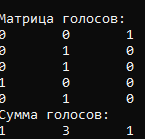
***Голосует избиратель №4(Вика):***

******

***Голосует избиратель №5(Максим):***

******

***Матрица и подсчёт голосов:***

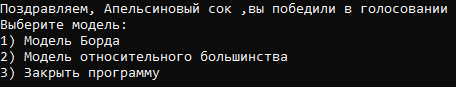
******

Яблочный сок-1

Апельсиновый сок-3

Вишнёвый сок-1

***Выбор победителя:***



**Приложение**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection;

namespace Votes

{

internal class Program

{

public static List<String> listCandidates;

public static List<String> listElectors;

static void Main(string[] args)

{

Console.WriteLine("Добро пожаловать!");

// Кандидаты

Console.WriteLine("Введите кол-во кандидатов для голосования:");

int candidates = Vvod(9);

Console.WriteLine();

// Список кандидатов

Console.WriteLine("Заполните список кандидатов:");

listCandidates = VvodList(candidates);

Console.WriteLine();

// Избиратели

Console.WriteLine("Введите кол-во избирателей для голосования:");

int electors = Vvod(9);

Console.WriteLine();

// Список избирателей

Console.WriteLine("Заполните список избирателей:");

listElectors = VvodList(electors);

Console.WriteLine();

// Использование модели

while (true)

{

// Выбор модели

Console.WriteLine("Выберите модель:");

Console.WriteLine("1) Модель Борда");

Console.WriteLine("2) Модель относительного большинства");

Console.WriteLine("3) Закрыть программу");

int model = Vvod(3);

Console.WriteLine();

switch (model)

{

case 1:

Bord();

break;

case 2:

Greater();

break;

case 3:

Console.WriteLine("До свидания!");

Environment.Exit(0);

break;

}

}

}

public static int Vvod(int max)

{

while (true)

{

string input = Console.ReadLine();

for (int i = 1; i <= max; i++)

{

if (input.Equals(Convert.ToString(i))) {

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("Значение сохранено!");

Console.ForegroundColor = ConsoleColor.White;

return Convert.ToInt16(input);

}

}

Console.WriteLine("Ошибка ввода, повторите попытку");

}

}

public static List<String> VvodList(int count)

{

List<String> list = new List<String>();

for (int i = 0; i < count; i++)

{

Console.WriteLine( (i+1));

list.Add(Console.ReadLine());

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("Значение сохранено");

Console.ForegroundColor = ConsoleColor.White;

}

return list;

}

public static void Bord()

{

Console.WriteLine("Проведём голосование!");

int[,] mass = new int[listElectors.Count, listCandidates.Count];

int[] sum = new int[listCandidates.Count];

// Опрос

for (int i = 0; i < listElectors.Count; i++)

{

Console.WriteLine("Голосует " + listElectors[i]);

for (int j = 0; j < listCandidates.Count; j ++)

{

Console.WriteLine("Выберите место для " + listCandidates[j] + " от {1 до " + listCandidates.Count + "}");

mass[i, j] = Vvod(listCandidates.Count);

}

}

Console.WriteLine();

// Вывод голосов

Console.WriteLine("Матрица голосов:");

for (int i = 0; i < listElectors.Count; i++)

{

for (int j = 0; j < listCandidates.Count; j++)

{

Console.Write(listCandidates.Count - mass[i, j] + "\t");

}

Console.WriteLine();

}

// Подсчёт голосов

for (int j = 0; j < listCandidates.Count; j++)

{

for (int i = 0; i < listElectors.Count; i++)

{

sum[j] += listCandidates.Count - mass[i, j];

}

}

// Вывод суммы баллов

Console.WriteLine("Сумма голосов:");

for (int i = 0; i < sum.Length; i++)

{

Console.Write(sum[i] + "\t");

}

Console.WriteLine();

int winnerId = Array.IndexOf(sum, sum.Max());

if (Array.FindAll(sum, value => value == sum.Max()).Length > 1)

Console.WriteLine("Невозможно определить победителя, потому что кол-во голосов совпадает");

else

Console.WriteLine("Поздравляем, " + listCandidates[winnerId] + " ,вы победили в голосовании");

}

public static void Greater()

{

Console.WriteLine("Проведём голосование!");

int[,] mass = new int[listElectors.Count, listCandidates.Count];

int[] sum = new int[listCandidates.Count];

// Опрос

for (int i = 0; i < listElectors.Count; i++)

{

Console.WriteLine("Голосует " + listElectors[i]);

Console.WriteLine("Проголосуйте за одного из кандидатов, от {1 до " + listCandidates.Count + "}");

int index = Vvod(listCandidates.Count);

mass[i, index-1] = 1;

}

Console.WriteLine();

// Вывод голосов

Console.WriteLine("Матрица голосов:");

for (int i = 0; i < listElectors.Count; i++)

{

for (int j = 0; j < listCandidates.Count; j++)

{

Console.Write(mass[i, j] + "\t");

}

Console.WriteLine();

}

// Подсчёт голосов

for (int j = 0; j < listCandidates.Count; j++)

{

for (int i = 0; i < listElectors.Count; i++)

{

sum[j] += mass[i, j];

}

}

// Вывод суммы баллов

Console.WriteLine("Сумма голосов:");

for (int i = 0; i < sum.Length; i++)

{

Console.Write(sum[i] + "\t");

}

Console.WriteLine();

int winnerId = Array.IndexOf(sum, sum.Max());

if (Array.FindAll(sum, value => value == sum.Max()).Length > 1)

Console.WriteLine("Невозможно определить победителя, потому что кол-во голосов совпадает");

else

Console.WriteLine("Поздравляем, " + listCandidates[winnerId] + " ,вы победили в голосовании");

}

}

}