|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

Студент\_\_\_\_\_Михалёв Ярослав Михайлович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_\_\_ИУ5-11Б\_\_\_\_\_\_\_\_\_\_\_\_

Тип практики ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ПРАКТИКА \_\_\_

Название предприятия КАФЕДРА «СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ» ( ИУ5 ) МГТУ им. Н.Э.Баумана

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**Михалёв Я.М.**\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**Козлов А.Д.**\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2021 г.*

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Написать протокол соревнований по гимнастике

У соревнования есть дата дд.мм.гг (проверить верность даты)

У каждого участника есть Фамилия и инициалы, дата рождения (лица моложе 16 не допускаются к соревнованиям (хотя бы +1 день) и старше 70 (хотя бы -1 день)), пол

Гимнастика:

Мужчины соревнуются в вольных упражнениях, на брусьях, в опорном прыжке и на кольцах

Женщины: в вольных упражнениях, на брусьях, в опорном прыжке и на бревне

Оценки целые от 0 до 10

Это одна запись о выступлении одного человека

Фильтрация участников с одинаковыми данными

Массив структур - протокол

Сортировка:

1. по сумме баллов

2. Дамы вперёд

3. Старые предпочтительнее

4. По алфавиту

ОГЛАВЛЕНИЕ

1. Введение
2. Техническое задание
3. Руководство пользователя
4. Исходный текст программы
5. Программа и методика испытаний
6. Список используемых источников

ВВЕДЕНИЕ

Целью и задачей данной проектно-технологической практики является освоение процедурного программирования на языке С++, углубление соответствующих навыков и дополнение теоретических знаний по дисциплине «Основы программирования», а также приобретение умения разрабатывать и оформлять техническую документацию на программный продукт.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

образования «Московский государственный технический университет имени

Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

|  |  |  |  |
| --- | --- | --- | --- |
|  | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |  | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_27\_» \_\_\_\_Декабря\_\_\_\_\_ 2021 г. |

**ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ПРАКТИКА**

**«Информационная система для сохранения данных участников соревнования»**

Техническое задание

4 листа

|  |
| --- |
| ИСПОЛНИТЕЛЬ:  Студент группы ИУ5-11Б Михалёв Я.М.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» 2021 г. |

Москва, 2021

**СОДЕРЖАНИЕ**

1. Наименование

2. Основание для разработки

3. Назначение разработки

4. Исполнитель

5. Технические требования

5.1. Требования к функциональным характеристикам

5.2. Требования к программному обеспечению

5.3. Требования к условиям эксплуатации

5.4. Требования к информационному обеспечению

5.5. Требования к надежности

5.6. Требования к составу и характеристикам технических средств

5.7. Требования к программной совместимости

5.8. Требования к маркеровке и упаковке программы

5.9. Требования к транспортировке и хранению

6. Требования к программной документации

7. Технико-экономические показатели

8. Стадии и этапы разработки

9. Порядок контроля и приемки задания

10. Дополнительные требования

# Наименование

Информационная система для сохранения данных участников соревнования.

# Основание для разработки

Основанием для разработки данного программного продукта является учебный план кафедры "Системы обработки информации и управления" МГТУ им. Н. Э. Баумана на 1-м семестре (дисциплина «Основы программирования»).

# Назначение разработки

Данный программный продукт предназначен для учета участников соревнования. В частности, программа должна обеспечивать решение задач: хранение данных программы в долговременной памяти, чтение данных из долговременной памяти в оперативную, просмотр данных о компьютерах организации с возможностью сортировки, добавление данных о новых компьютерах, удаление данных о компьютере, изменение данных о компьютере. Программа должна обеспечивать удобную работу с данными, высокий уровень надежности и функциональных возможностей.

# Исполнитель

Студент группы ИУ5-11Б МГТУ им. Баумана Михалёв Ярослав Михайлович, вариант № 11.

# Технические требования

## 5.1. Требования к функциональным характеристикам

**5.1.1. Общие требования**

5.1.1.1. Интерфейс программы должен быть реализован в консоли.

5.1.1.2. Интерфейс программы должен быть полностью на русском языке, а также максимально интуитивным и понятным, каждое действие должно сопровождаться пояснением.

5.1.1.3. Ниже по тексту, данные о одном спортсмене называются записью.

5.1.1.4. Интерфейс программы должен быть реализован в виде всплывающего вопроса.

5.1.1.5. Программа должна игнорировать некорректно введенные данные и сообщать об ошибке пользователю.

5.1.1.6. При выходе из программы осуществляется завершение программы без сохранения данных.

**5.1.2. Основные функции.**

Программа должна обеспечивать выполнение следующих функций:

- Добавление данных о участнике соревнования

5.1.2.1. **Просмотр данных**.

Вывод данных в консоль должен быть отформатирован и отсортирован.

5.1.2.2. **Добавление данных**.

При добавлении данных формат вводимых данных должен быть в следующем виде: фамилия и инициалы, дата рождения, пол, балл за каждую дисциплину.

При добавлении данных о спортсмене, имя которого уже содержится в списке, необходимо предоставить пользователю возможность выбрать запись: вновь введенную или ту, которая уже есть в списке.

5.1.2.3. **Сортировка данных**.

5.1.2.3.1. В программе должна быть реализована сортировка по убыванию количества баллов.

5.1.2.3.2. В программе должна быть реализована сортировка по гендерной принадлежности участника соревнования.

5.1.2.3.3. В программе должна быть реализована сортировка по возрасту участника (от старших к младшим).

5.1.2.3.4. В программе должна быть реализована сортировка участников по алфавиту.

## 5.2. Требования к программному обеспечению

5.2.1. Данная программа предназначена для выполнения на компьютере под управлением системы Microsoft Windows версии XP и выше. Для использования данной программы требуется наличие на компьюторе компилятора С++ и среды программирования Microsoft Visual Studio 2019 (MS VS).

## 5.3. Требования к условиям эксплуатации

5.3.1. Данная программа должна эксплуатироваться на компьюторе под управлением системы Microsoft Windows версии XP и выше без установленной среды программирования MS VS.

5.3.2. Для сопровождения программы системный программист должен быть знаком с языком C++ в среде программирования MS VS 2019. Для работы с данной программой системный программист должен быть знаком с навыками объектно-ориентированного программирования.

## 5.4. Требования к информационному обеспечению

5.4.1. Программа должна быть реализована на языке С++.

5.4.2. Информация о структурах данных программы должна быть представлена в техническом описании.

5.4.3. Программа должна быть представлена набором файлом с расширениями .cpp и .h.

5.4.4. Ввод и вывод в программе должен быть на русском языке.

5.4.5. Комментарии в листинге программы должны быть на русском языке.

5.4.6. Программа должна работать в консольном режиме.

## 5.5. Требования к надежности

5.5.1. Программа должна безотказно работать при отсутствии сбоев операционной системы и технического обеспечения.

5.5.2. В программе не допускается выделение динамической памяти без последующего её освобождения.

## 5.6. Требования к составу и характеристикам технических средств

Данная программа должна выполняться на компьютерах следующей конфигурации:

5.6.1. Компьютер с процессором Intel 80486 и выше;

5.6.2. Не менее 2гб свободной оперативной памяти;

5.6.3. Монитор с VGA выходом;

5.6.4. Стандартная клавиатура;

5.6.5. Свободное место на жестком диске не менее 450 Кбайт.

## 5.7. Требования к программной совместимости

Для сопровождения программы должна использоваться программа Visual Studio 2019, компьютер должен быть оснащен русской таблицей символов и кодовой поддержкой.

## 5.8. Требования к маркеровке и упаковке программы

Программа предоставляется на флэш-носителе или другом сменном носителе информации.

## 5.9. Требования к транспортировке и хранению

Программа транспортируется и хранится на флэш-носителе или другом сменном носителе информации.

# Требования к программной документации

6.1. По окончанию работы должны быть предъявлены следующие документы:

6.1.1. Техническое задание;

6.1.2. Программа и методика испытаний;

6.1.3. Руководство пользователя;

6.1.4. Исходный текст программы.

# 7. Технико-экономические показатели

7.1. Требования по данным показателям при выполнении задания проектно-технологической практики не предъявляются.

# 

# 8. Стадии и этапы разработки

8.1. Сроки выполнения отдельных этапов работ

8.1.1. Получение и уточнение задания – 12 неделя семестра.

8.1.2. Подписание ТЗ – 13 неделя семестра.

8.1.3. Разработка программы – 14 неделя семестра.

8.1.4. Кодирование и отладка программы – 15 неделя семестра.

8.1.5. Разработка документации – 16 неделя семестра.

8.1.6. Защита и проведение испытаний – 17 неделя семестра.

# 9. Порядок контроля и приемки задания

9.1. Тестирование программного продукта будет осуществляться в соответствии с документом «Программа и методика испытаний» (ПМИ) на компьютере, который удовлетворяет пункту 5.6.1.

9.2. Контроль и приемка программы и программной документации производится дистанционно в программе Discord.

9.3. Для проверки и тестирования программы предоставляется исполняемый модуль (\*.exe) и комплект файлов для сборки в VS 2019.

# 10. Дополнительные требования

Данное техническое задание может уточняться в процессе разработки.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

образования «Московский государственный технический университет имени

Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

|  |  |  |  |
| --- | --- | --- | --- |
|  | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Правдина  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

**ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ПРАКТИКА**

**«Информационная система для сохранения данных участников соревнования»**

Руководство пользователя

6 листов

|  |
| --- |
| ИСПОЛНИТЕЛЬ:  Студент группы ИУ5-11Б Михалёв Я.М.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» 2021 г. |

Москва, 2021

**СОДЕРЖАНИЕ**

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

2. ПОДКЛЮЧЕНИЕ И ЗАПУСК ПРОГРАММЫ

3. РАБОТА С ПРОГРАММОЙ

3.1. Добавление новых данных в список

3.2 Сортировка данных

3.3 Выход из программы

4. ОПИСАНИЕ СООБЩЕНИЙ ОБ ОШИБКАХ И ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЯХ ПРИ РАБОТЕ С ПРОГРАММОЙ

4.1. Неправильный формат ввода

## 4.2. Участник с идентичным именем уже существует

# НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

Целью работы является создание программы для автоматизации учета данных участников соревнования. В частности, программа обеспечивает решение следующих задач: данных о каждом участнике; просмотр данных о каждом участнике с возможностью сортировки; добавление данных о новых участниках. Программа обеспечивает удобную работу с данными, высокий уровень надежности и функциональных возможностей.

# ПОДКЛЮЧЕНИЕ и запуск ПРОГРАММЫ

Для установки данной программы на компьютер следует выполнить следующие действия:

2.1. Освободить на жестком диске как минимум 450 килобайт свободного пространства.

2.2. Вставить флеш-носитель (или другой сменный носитель информации) с исходным модулем программы в USB-порт.

2.3. Скопировать файл программы, который указан в разделе 4 этого документа, с носителя в нужный каталог.

# 3. РАБОТА c ПРОГРАММой

После первого запуска программы в консоли появляется предложение (рис.1).

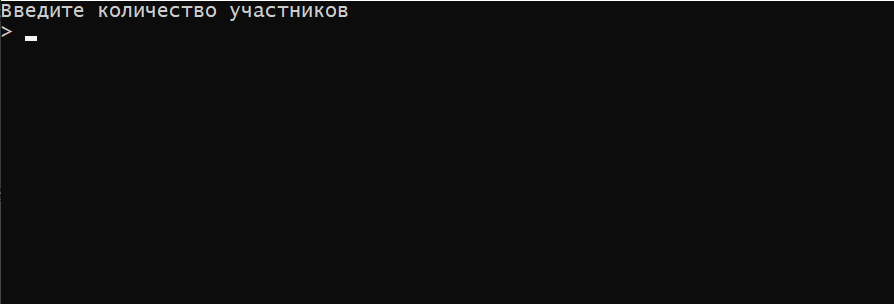


Рис.1.

## 3.1. ДОБАВЛЕНИЕ НОВЫХ ДАННЫХ В СПИСОК

Введите количество участников. Далее через Enter нужно ввести данные о каждом участнике соревнования в следующем порядке: имя участника, день рождения участника, месяц рождения участника, год рождения участника, день проведения соревнований, месяц проведения соревнований, год проведения соревнований, пол участника, количество очков по каждой дисциплине, суммарное количество очков (рис. 2).

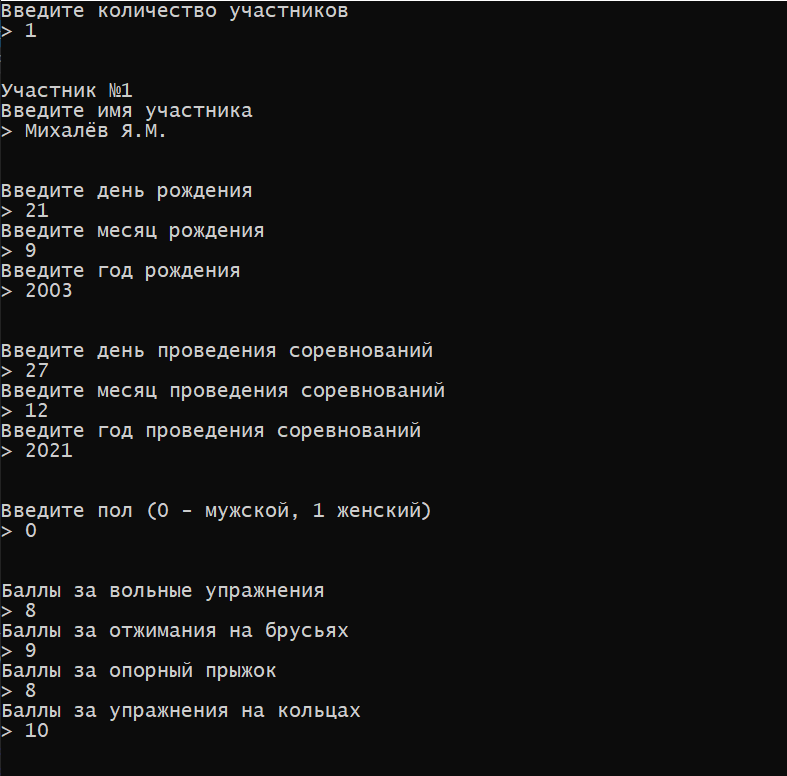


Рис.2.

# 3.2. СОРТИРОВКА ДАННЫХ

Сортировка данных осуществляется по суммарному баллу, полу участника, возрасту и алфавитному порядку (Рис. 3).

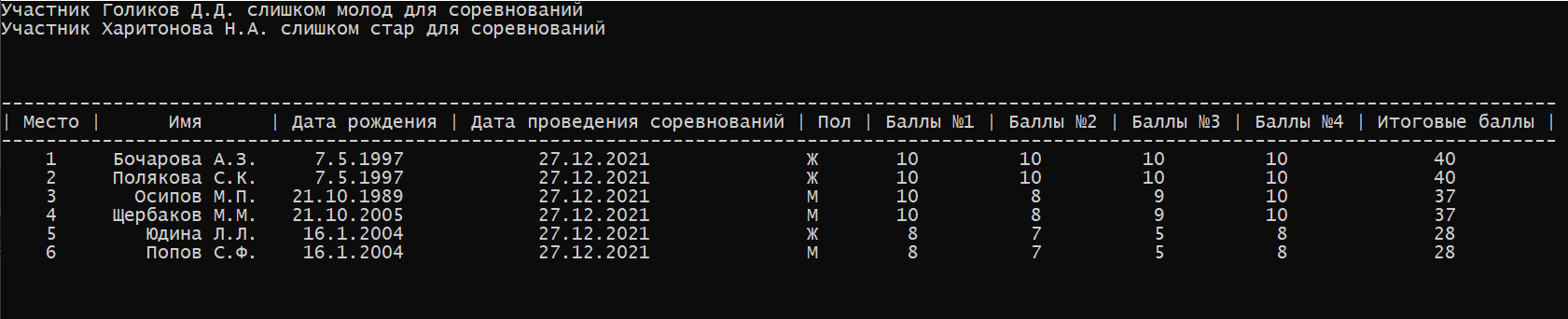


Рис.3.

# 3.3. вЫХОД ИЗ ПРОГРАММЫ

Для выхода из программы достаточно нажать любую клавишу.

# 

# 4. ОписАНИЕ СООБЩЕНИЙ ОБ ОШИБКАХ И ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦиях ПРИ РАБОТЕ С ПРОГРАММОЙ

## 4.1. Неправильный формат ввода

Если пользователь введет неправильные данные, то программа выведет предупреждение и попросит заново ввести данные (Рис. 4-7).

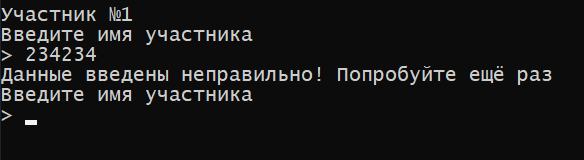


Рис. 4.

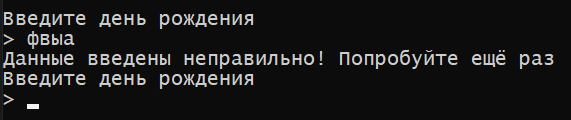


Рис. 5.

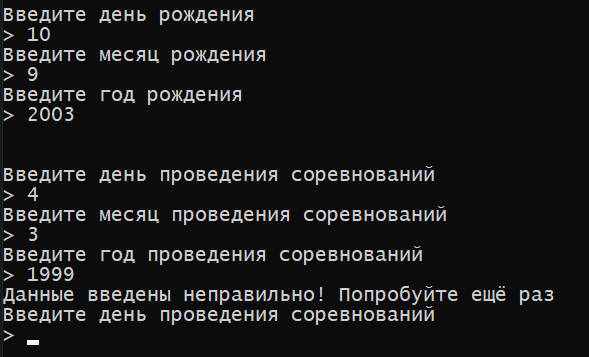


Рис. 6.

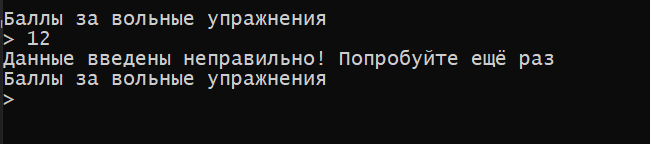


Рис. 7.

**Имя участника:** допускаются все буквы английского и русского алфавита, прописные и строчные.

**День рождения участника:** допускаются натуральные числа, от 1 до 31.

**Месяц рождения участника:** допускаются натуральные числа, от 1 до 12.

**Год рождения участника:** допускаются натуральные числа > 1970.

## 4.2. Участник с идентичным именем уже существует

Если вы встретили данное сообщение при работе с программой, то имя участника в только что введенной записи совпадает с именем участника в уже имеющихся записях. Вам будет предложено заново ввести имя участника (рис. 8)

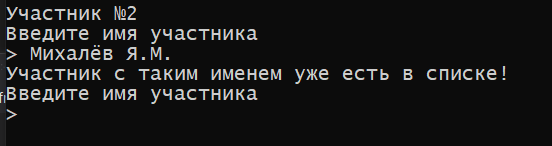


Рис. 8

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

образования «Московский государственный технический университет имени

Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

|  |  |  |
| --- | --- | --- |
|  | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

**ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ПРАКТИКА**

**«Информационная система для сохранения данных участников соревнования»**

Исходный текст программы

8 листов

|  |
| --- |
| ИСПОЛНИТЕЛЬ:  Студент группы ИУ5-11Б Михалёв Я.М.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» 2021 г. |

Москва, 2021

**Содержимое**

Файл Header.h

#include <string>

**using** **namespace** std**;**

#pragma once

int GetConsoleWidth**();**

void h1**(**const char**\*** s**);**

int Sum**(**int arr**[],** int n **=** 4**);**

bool isString**(**string s**);**

bool CheckDate**(**int date**[**3**]);**

double difference**(**int bDay**[**3**],** int cDay**[**3**]);**

string ShowGender**(**int gender**);**

string FormatDate**(**int date**[**3**]);**

bool isDuplicate**(**string name**);**

int Input**(**string message**,** int left **=** 0**,** int right **=** 9999**);**

string GetName**();**

void GetBirthdayDate**(**int birthday**[**3**]);**

void GetСompetitionDate**(**int birthday**[**3**],** int competition**[**3**]);**

int GetGender**();**

void GetScore**(**int arr**[**4**],** int gender**);**

Файл main.cpp

#include <iostream>

#include <iomanip>

#include <string>

#include <vector>

#include <algorithm>

#include <Windows.h>

#include "Header.h"

**using** **namespace** std**;**

struct Contest **{**

string name**;**

int bDate**[**3**];**

int cDate**[**3**];**

int gender**;**

int score**[**4**];**

int sum**;**

**};**

vector**<**Contest**>** table**;**

// Возращает true, если в спике уже есть участник с именем name, в обратном случае возращает false

bool isDuplicate**(**string name**)**

**{**

**for** **(**int i **=** 0**;** i **<** table**.**size**();** i**++)**

**if** **(**table**[**i**].**name **==** name**)**

**return** **true;**

**return** **false;**

**}**

// Добавляет нового участника в список

void NewMember**(**string name**,** int bDate\_day**,** int bDate\_month**,** int bDate\_year**,** int cDate\_day**,** int cDate\_month**,** int cDate\_year**,** int gender**,** int score0**,** int score1**,** int score2**,** int score3**)**

**{**

table**.**push\_back**(**Contest**());**

int i **=** table**.**size**()** **-** 1**;**

table**[**i**].**name **=** name**;**

table**[**i**].**bDate**[**0**]** **=** bDate\_day**;**

table**[**i**].**bDate**[**1**]** **=** bDate\_month**;**

table**[**i**].**bDate**[**2**]** **=** bDate\_year**;**

table**[**i**].**cDate**[**0**]** **=** cDate\_day**;**

table**[**i**].**cDate**[**1**]** **=** cDate\_month**;**

table**[**i**].**cDate**[**2**]** **=** cDate\_year**;**

table**[**i**].**gender **=** gender**;**

table**[**i**].**score**[**0**]** **=** score0**;**

table**[**i**].**score**[**1**]** **=** score1**;**

table**[**i**].**score**[**2**]** **=** score2**;**

table**[**i**].**score**[**3**]** **=** score3**;**

table**[**i**].**sum **=** score0 **+** score1 **+** score2 **+** score3**;**

**}**

// Добавляет n участников в список

void AddMembers**(**int n **=** 1**)**

**{**

string name**;**

int bDate**[**3**];**

int cDate**[**3**];**

int gender**;**

int score**[**4**];**

int sum**;**

**for** **(**int i **=** 0**;** i **<** n**;** i**++)** **{**

table**.**push\_back**(**Contest**());**

int j **=** table**.**size**()** **-** 1**;**

cout **<<** "\n\nУчастник №" **<<** j **+** 1 **<<** endl**;**

name **=** GetName**();**

GetBirthdayDate**(**bDate**);**

GetСompetitionDate**(**bDate**,** cDate**);**

gender **=** GetGender**();**

GetScore**(**score**,** gender**);**

table**[**j**].**name **=** name**;**

table**[**j**].**bDate**[**0**]** **=** bDate**[**0**];**

table**[**j**].**bDate**[**1**]** **=** bDate**[**1**];**

table**[**j**].**bDate**[**2**]** **=** bDate**[**2**];**

table**[**j**].**cDate**[**0**]** **=** cDate**[**0**];**

table**[**j**].**cDate**[**1**]** **=** cDate**[**1**];**

table**[**j**].**cDate**[**2**]** **=** cDate**[**2**];**

table**[**j**].**gender **=** gender**;**

table**[**j**].**score**[**0**]** **=** score**[**0**];**

table**[**j**].**score**[**1**]** **=** score**[**1**];**

table**[**j**].**score**[**2**]** **=** score**[**2**];**

table**[**j**].**score**[**3**]** **=** score**[**3**];**

table**[**j**].**sum **=** Sum**(**score**,** 4**);**

**}**

**}**

// Фильтрует список участников по возрасту

void AgeFilter**()**

**{**

**for** **(**int i **=** 0**;** i **<** table**.**size**();** i**++)**

**{**

int age **=** difference**(**table**[**i**].**bDate**,** table**[**i**].**cDate**)** **/** 365**;**

**if** **(**age **<** 16**)**

**{**

cout **<<** "Участник " **<<** table**[**i**].**name **<<** " слишком молод для соревнований" **<<** endl**;**

table**.**erase**(**table**.**begin**()** **+** i**);**

i**--;**

**}**

**else** **if** **(**age **>** 70**)**

**{**

cout **<<** "Участник " **<<** table**[**i**].**name **<<** " слишком стар для соревнований" **<<** endl**;**

table**.**erase**(**table**.**begin**()** **+** i**);**

i**--;**

**}**

**}**

cout **<<** "\n\n\n"**;**

**}**

// Сортирует список участников по набранным баллам, полу, возрасту и алфавиту

void SortMembers**()**

**{**

**for** **(**int i **=** 0**;** i **<** table**.**size**()** **-** 1**;** i**++)**

**for** **(**int j **=** 0**;** j **<** table**.**size**()** **-** i **-** 1**;** j**++)**

**{**

**if** **(**table**[**j**].**sum **<** table**[**j **+** 1**].**sum**)**

swap**(**table**[**j**],** table**[**j **+** 1**]);**

**else** **if** **(**table**[**j**].**sum **==** table**[**j **+** 1**].**sum**)**

**{**

**if** **(**table**[**j**].**gender **<** table**[**j **+** 1**].**gender**)**

swap**(**table**[**j**],** table**[**j **+** 1**]);**

**else** **if** **(**table**[**j**].**gender **==** table**[**j **+** 1**].**gender**)**

**{**

**if** **(**difference**(**table**[**j**].**bDate**,** table**[**j **+** 1**].**bDate**)** **<** 0**)**

swap**(**table**[**j**],** table**[**j **+** 1**]);**

**else** **if** **(**difference**(**table**[**j**].**bDate**,** table**[**j **+** 1**].**bDate**)** **==** 0**)**

**{**

**if** **(**table**[**j**].**name **>** table**[**j **+** 1**].**name**)**

swap**(**table**[**j**],** table**[**j **+** 1**]);**

**}**

**}**

**}**

**}**

**}**

// Выводит таблицу участников соревнований

void ShowMembers**()**

**{**

h1**(**"| Место | Имя | Дата рождения | Дата проведения соревнований | Пол | Баллы №1 | Баллы №2 | Баллы №3 | Баллы №4 | Итоговые баллы |"**);**

**for** **(**int i **=** 0**;** i **<** table**.**size**();** i**++)** **{**

cout **<<** setw**(**5**)** **<<** i **+** 1 **<<** setw**(**5**);**

cout**.**width**(**18**);**

cout **<<** right **<<** table**[**i**].**name**;**

cout**.**width**(**13**);**

cout **<<** FormatDate**(**table**[**i**].**bDate**);**

cout**.**width**(**22**);**

cout **<<** FormatDate**(**table**[**i**].**cDate**);**

cout**.**width**(**15**);**

cout **<<** ShowGender**(**table**[**i**].**gender**);**

cout**.**width**(**9**);**

cout **<<** table**[**i**].**score**[**0**];**

cout**.**width**(**11**);**

cout **<<** table**[**i**].**score**[**1**];**

cout**.**width**(**11**);**

cout **<<** table**[**i**].**score**[**2**];**

cout**.**width**(**11**);**

cout **<<** table**[**i**].**score**[**3**];**

cout**.**width**(**15**);**

cout **<<** table**[**i**].**sum**;**

cout **<<** "\n"**;**

**}**

cout **<<** "\n\n\n"**;**

**}**

int main**()**

**{**

setlocale**(**LC\_ALL**,** "RUS"**);**

SetConsoleCP**(**1251**);**

SetConsoleOutputCP**(**1251**);**

NewMember**(**"Голиков Д.Д."**,** 13**,** 5**,** 2010**,** 27**,** 12**,** 2021**,** 0**,** 4**,** 5**,** 7**,** 6**);**

NewMember**(**"Харитонова Н.А."**,** 13**,** 5**,** 1980**,** 27**,** 12**,** 2100**,** 1**,** 6**,** 7**,** 6**,** 6**);**

NewMember**(**"Попов С.Ф."**,** 16**,** 1**,** 2004**,** 27**,** 12**,** 2021**,** 0**,** 8**,** 7**,** 5**,** 8**);**

NewMember**(**"Юдина Л.Л."**,** 16**,** 1**,** 2004**,** 27**,** 12**,** 2021**,** 1**,** 8**,** 7**,** 5**,** 8**);**

NewMember**(**"Бочарова А.З."**,** 7**,** 5**,** 1997**,** 27**,** 12**,** 2021**,** 1**,** 10**,** 10**,** 10**,** 10**);**

NewMember**(**"Полякова С.К."**,** 7**,** 5**,** 1997**,** 27**,** 12**,** 2021**,** 1**,** 10**,** 10**,** 10**,** 10**);**

NewMember**(**"Осипов М.П."**,** 21**,** 10**,** 1989**,** 27**,** 12**,** 2021**,** 0**,** 10**,** 8**,** 9**,** 10**);**

NewMember**(**"Щербаков М.М."**,** 21**,** 10**,** 2005**,** 27**,** 12**,** 2021**,** 0**,** 10**,** 8**,** 9**,** 10**);**

int n**;**

cout **<<** "Введите количество участников\n> "**;**

cin **>>** n**;**

AddMembers**(**n**);**

AgeFilter**();**

SortMembers**();**

ShowMembers**();**

**}**

utils.cpp

#include <iostream>

#include <Windows.h>

#include <string>

#include <algorithm>

#include "Header.h"

#define ERROR\_MESSAGE "Данные введены неправильно! Попробуйте ещё раз"

**using** **namespace** std**;**

// Запрашивает число на отрезке [left, right]

int Input**(**string message**,** int left**,** int right**)**

**{**

int tmp**;**

**while** **(true)**

**{**

cout **<<** message **<<** "\n> "**;**

cin **>>** tmp**;**

**if** **(!**cin**.**fail**()** **&&** **(**tmp **>=** left **&&** tmp **<=** right**))**

**{**

**break;**

**}**

cout **<<** ERROR\_MESSAGE **<<** endl**;**

cin**.**clear**();**

cin**.**ignore**(**1000**,** '\n'**);**

**}**

**return** tmp**;**

**}**

// Запрашивает имя участника

string GetName**()**

**{**

string name**;**

cin**.**ignore**(**1000**,** '\n'**);**

**while** **(true)**

**{**

cout **<<** "Введите имя участника\n> "**;**

getline**(**cin**,** name**);**

**if** **(**isString**(**name**))**

**if** **(!**isDuplicate**(**name**))**

**break;**

**else**

cout **<<** "Участник с таким именем уже есть в списке!" **<<** endl**;**

**else**

cout **<<** ERROR\_MESSAGE **<<** endl**;**

**}**

cout **<<** "\n\n"**;**

**return** name**;**

**}**

// Запрашивает дату рождения участника

void GetBirthdayDate**(**int birthday**[**3**])**

**{**

**while** **(true)**

**{**

birthday**[**0**]** **=** Input**(**"Введите день рождения"**,** 1**,** 31**);**

birthday**[**1**]** **=** Input**(**"Введите месяц рождения"**,** 1**,** 12**);**

birthday**[**2**]** **=** Input**(**"Введите год рождения"**);**

**if** **(**CheckDate**(**birthday**))**

**break;**

**else**

cout **<<** ERROR\_MESSAGE **<<** endl**;**

**}**

cout **<<** "\n\n"**;**

**}**

// Запрашивает дату соревнования

void GetСompetitionDate**(**int birthday**[**3**],** int competition**[**3**])**

**{**

**while** **(true)**

**{**

competition**[**0**]** **=** Input**(**"Введите день проведения соревнований"**,** 1**,** 31**);**

competition**[**1**]** **=** Input**(**"Введите месяц проведения соревнований"**,** 1**,** 12**);**

competition**[**2**]** **=** Input**(**"Введите год проведения соревнований"**);**

**if** **(**CheckDate**(**competition**)** **&&** difference**(**birthday**,** competition**)** **>** 0**)**

**break;**

**else**

cout **<<** ERROR\_MESSAGE **<<** endl**;**

**}**

cout **<<** "\n\n"**;**

**}**

// Запрашивает пол

int GetGender**()**

**{**

int gender **=** Input**(**"Введите пол (0 - мужской, 1 женский)"**,** 0**,** 1**);**

cout **<<** "\n\n"**;**

**return** gender**;**

**}**

// Запрашивает количество очков за каждую дисциплину

void GetScore**(**int arr**[**4**],** int gender**)**

**{**

arr**[**0**]** **=** Input**(**"Баллы за вольные упражнения"**,** 0**,** 10**);**

arr**[**1**]** **=** Input**(**"Баллы за отжимания на брусьях"**,** 0**,** 10**);**

arr**[**2**]** **=** Input**(**"Баллы за опорный прыжок"**,** 0**,** 10**);**

**if** **(**gender **==** 0**)**

// Если участник - мужчина

arr**[**3**]** **=** Input**(**"Баллы за упражнения на кольцах"**,** 0**,** 10**);**

**else**

// Если участник - женщина

arr**[**3**]** **=** Input**(**"Баллы за упражнения на бревне"**,** 0**,** 10**);**

**}**

// Вовращает ширину консоли

int GetConsoleWidth**()**

**{**

COORD**\*** Chars**;**

HANDLE hCon **=** GetStdHandle**(**STD\_OUTPUT\_HANDLE**);**

CONSOLE\_SCREEN\_BUFFER\_INFO csbInfo**;**

GetConsoleScreenBufferInfo**(**hCon**,** **&**csbInfo**);**

int width **=** csbInfo**.**srWindow**.**Right **-** csbInfo**.**srWindow**.**Left**;**

**return** width**;**

**}**

// Перезагружает оператор \*

string **operator** **\*** **(**string a**,** int b**)** **{**

string output **=** ""**;**

**while** **(**b**--)**

output **+=** a**;**

**return** output**;**

**}**

// Выводит сообщение на экран

void h1**(**const char**\*** s**)**

**{**

int width **=** GetConsoleWidth**();**

cout **<<** string**(**"-"**)** **\*** width **<<** endl**;**

cout **<<** s **<<** endl**;**

cout **<<** string**(**"-"**)** **\*** width **<<** endl**;**

**}**

// Возращает сумму элементов массива arr размером n

int Sum**(**int arr**[],** int n**)**

**{**

int res **=** 0**;**

**for** **(**int i **=** 0**;** i **<** n**;** i**++)**

res **+=** arr**[**i**];**

**return** res**;**

**}**

// Возращает true, если строка полностью состоит из букв, false в обратном случае

bool isString**(**string s**)**

**{**

**return** **!**std**::**any\_of**(**s**.**begin**(),** s**.**end**(),** **::**isdigit**);**

**}**

// Проверяет дату на корректность

bool CheckDate**(**int date**[**3**])**

**{**

struct tm a **=** **{** 0**,** 0**,** 0**,** date**[**0**],** date**[**1**]** **-** 1**,** date**[**2**]** **-** 1900 **};**

time\_t x **=** mktime**(&**a**);**

**return** x **!=** **(**time\_t**)(-**1**);**

**}**

// Возращает разницу в днях между двумя датами

double difference**(**int bDay**[**3**],** int cDay**[**3**])**

**{**

struct tm a **=** **{** 0**,** 0**,** 0**,** bDay**[**0**],** bDay**[**1**]** **-** 1**,** bDay**[**2**]** **-** 1900 **};**

struct tm b **=** **{** 0**,** 0**,** 0**,** cDay**[**0**],** cDay**[**1**]** **-** 1**,** cDay**[**2**]** **-** 1900 **};**

time\_t x **=** mktime**(&**a**);**

time\_t y **=** mktime**(&**b**);**

**if** **(**x **!=** **(**time\_t**)(-**1**)** **&&** y **!=** **(**time\_t**)(-**1**))**

**return** difftime**(**y**,** x**)** **/** **(**60 **\*** 60 **\*** 24**);**

**return** 0**;**

**}**

// Возращает пол

string ShowGender**(**int gender**)**

**{**

**if** **(**gender **==** 0**)**

**return** "М"**;**

**else**

**return** "Ж"**;**

**}**

// Возращает дату в формате дд.мм.гг

string FormatDate**(**int date**[**3**])**

**{**

**return** to\_string**(**date**[**0**])** **+** "." **+** to\_string**(**date**[**1**])** **+** "." **+** to\_string**(**date**[**2**]);**

**}**

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

образования «Московский государственный технический университет имени

Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

|  |  |  |
| --- | --- | --- |
|  | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Козлов  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

**ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ПРАКТИКА**

**«Информационная система для сохранения данных участников соревнования»**

Программа и методика испытаний

3 листа

|  |
| --- |
| ИСПОЛНИТЕЛЬ:  Студент группы ИУ5-11Б Михалёв Я.М.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» 2021 г. |

Москва, 2021

**СОДЕРЖАНИЕ**

1. ОБЪЕКТ ИСПЫТАНИЙ

2. ЦЕЛЬ ИСПЫТАНИЙ

3. СОСТАВ ПРЕДЪЯВЛЯЕМОЙ ДОКУМЕНТАЦИИ

4. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ИСПЫТАНИЯМ И УСЛОВИЯ

ПРОВЕДЕНИЯ ИСПЫТАНИЯ

4.1. Требования к условиям проведения испытаний

4.2. Требования к техническим средствам

5. ПОРЯДОК ПРОВЕДЕНИЯ ИСПЫТАНИЯ

5.1. Состав и структура технических и программных средств

5.2. Последовательность испытаний программы

6. РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

# ОБЪЕКТ ИСПЫТАНИЙ

Объектом испытаний является программа для хранения данных участников соревнований.

# ЦЕЛЬ ИСПЫТАНИЙ

Целью испытаний программного продукта состоит в проверке работоспособности программы и проверке соответствия основных функциональных возможностей программы требованиям, заложенным в техническом задании на программный продукт.

# СОСТАВ ПРЕДЪЯВЛЯЕМОЙ ДОКУМЕНТАЦИИ

При проведении испытаний предъявляются следующие документы:

* 1. Техническое задание (ТЗ)
  2. Руководство пользователя (РП)
  3. Исходный текст программы
  4. Программа и методика испытаний (ПМИ)

# ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К ИСПЫТАНИЯМ И УСЛОВИЯ ПРОВЕДЕНИЯ ИСПЫТАНИЯ

## 4.1. Требования к условиям проведения испытаний

4.1.1. Программа должна работать в операционной среде, где не инсталлирована система программирования (MS VS).

4.1.2. Программа должна работать в режиме эмуляции командной строки (cmd.exe), при этом вывод и ввод информации должен быть обеспечен на русском языке.

4.1.3. При необходимости и наличии системы программирования (MS VS), сборка проекта программы должна быть выполнена без ошибок и предупреждений.

## 4.2. Требования к техническим средствам

4.2.1. Компьютер с процессором Intel 80486 и выше

4.2.2. 2 гигабайта оперативной памяти

4.2.3. Монитор с VGA выходом

4.2.4. Стандартная клавиатура

4.2.5. 42 килобайта свободного места на жестком диске для программы

4.2.6. 400 килобайт свободного места на жестком диске для базы данных.

# 5. ПОРЯДОК ПРОВЕДЕНИЯ ИСПЫТАНИЯ

## 5.1. Состав и структура технических и программных средств

5.1.1. Запуск программы выполняется в режиме командной строки (cmd.exe) или из любого файл-менеджера.

5.1.2. Запуск режима командной строки производиться из главного меню ОС: ПУСК-> ВЫПОЛНИТЬ-> cmd.exe.

5.1.3. Программа для испытаний запускается с флеш-носителя или другого сменного носителя информации. Для запуска программы из текущей директории вводится: IISAWACC.exe.

## 5.2. Последовательность испытаний программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **№ пункта Т.З.** | **Выполняемые действия** | **Ожидаемый результат** | **Примечание** |
|  | Запуск программы |  | Выведется надпись (Рис. 1) | Включает пункты 5.1.1.2, 5.1.1.3, 5.1.1.4 |
|  | 5.1.2.1 и 5.1.2.2  Добавление нескольких записей | Ввод данных согласно всплывающим надписям | Выведется надпись  (Рис. 2-7) | Данные водятся через Enter, Включает пункт 5.1.1.4 |
|  | 5.1.1.5  Неверно введенные данные | Ввести такую дату рождения участника, чтобы разница между датой рождения и датой соревнования составляла меньше 17 лет или больше 70 лет. | Выведется надпись и потребуется новый ввод данных участника (Рис. 8) | Включает пункт 5.1.1.4 |
|  | 5.1.1.6 Завершение программы | После вывода данных на экран нажать Enter | Завершение программы |  |

# 6. РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Рис. 9

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации (ЕСПД)
2. Эккель Б. Философия C++. Введение в стандартный C++. 2-е изд. — СПб.: Питер, 2004. — 572 с.: ил.Г. Шилдт “С++ Базовый курс”: Пер. с англ.- М., Издательский дом “Вильямс”, 2011 г. – 672с
3. Шилдт Г., “С++ Руководство для начинающих”: Пер. с англ. - М., Издательский дом “Вильямс”, 2005 г. – 672с
4. Шилдт Г., “Полный справочник по С++”: Пер. с англ.- М., Издательский дом “Вильямс”, 2006 г. – 800с
5. Страуструп Б., "Язык программирования С++"- М., Бином, 2010 г.
6. MSDN Library for Visual Studio 2005 (Microsoft Document Explorer – входить в состав дистрибутива VS. Нужно обязательно развернуть при установке MS VS!)
7. Керниган Б., Ритчи Д. К36 Язык программирования Си.\Пер. с англ., 3-е изд., испр. - СПб.: "Невский Диалект", 2001. - 352 с.: ил.
8. Страуструп Б. "Дизайн и эволюция С++. Классика CS" – СПб.: Питер, 2007. – 445с.