

Спецификация конкурсных материалов для проведения практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации *Инженерный класс* по направлению программирование

1. Назначение конкурсных материалов

Материалы *практического* этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) предназначены для оценки уровня *практической* подготовки участников Конкурса.

2. Условия проведения

Практический этап Конкурса проводится в *очной форме на базе вуза*. При выполнении работы обеспечивается строгое соблюдение порядка организации и проведения Конкурса.

3. Продолжительность выполнения

На выполнение заданий *практического* этапа Конкурса отводится *90 минут*.

4. Содержание и структура

Задания *практического* этапа Конкурса разработаны преподавателями образовательных организаций высшего образования, участвующих в проекте “*Инженерный класс* в московской школе”.

Индивидуальный вариант участника включает *1 кейса*, базирующихся на содержании одного из *элективных курсов* “*Микроэлектроника и схемотехника*” и “*Основы программирования на C++*”, “*Основы программирования на Python*”, “*Язык программирования Java*”. Тему кейса выбирает учащийся перед сдачей *практического* этапа.

5. Система оценивания

Задание считается выполненным, если ответ участника выполнил задание, поставленное в условиях кейса и соответствует требованиям задания. Каждый кейс оценивается в *60 баллов*. Максимальный балл за выполнение всех заданий – *60 баллов*. Для получения максимального балла за *практический* этап Конкурса необходимо дать верные ответы на все задания.

6. Приложения

1. Обобщённый план конкурсных материалов для проведения *практического* этапа Конкурса.
2. Демонстрационный вариант конкурсных заданий *практического* этапа Конкурса.

Обобщённый план конкурсных материалов для проведения *практического* этапа Конкурса

| № задания | Уровень сложности | <i>Элективный курс</i> | Контролируемые требования к проверяемым умениям | Балл |
|-----------|-------------------|--|---|------|
| | <i>повышенный</i> | <i>Микроэлектроника и схемотехника</i> | Создание схемы устройства и его программирование на языке программирования по описанию устройства и перечисленным элементам. | 60 |
| | <i>повышенный</i> | <i>Основы программирования на C++</i> | Основные элементы языка, основные алгоритмические конструкции (циклы, условия) Обработка строк Обработка одномерных и двумерных массивов Чтение данных из файла, запись данных в файл. | 60 |
| | <i>повышенный</i> | <i>Основы программирования на Python</i> | Типы данных и операции Инструкции и синтаксис Структуры и функции в Python | 60 |
| | <i>повышенный</i> | <i>Язык программирования Java</i> | Типы данных, синтаксис и методы в Java. | 60 |
| | | | | |

Демонстрационный вариант конкурсных заданий *практического* этапа Конкурса**Кейс №1. Курс “Микроэлектроника и схемотехника”**

Задание:

Собрать схему светофора с переключением на разрешающий свет пешеходам по приближению пешехода на расстояние меньше 1-го метра. Установить время, равное 15-ти секундам на работу разрешающего света для пешеходов. Переключение светофора на зеленый свет для автомобилей производить автоматически по истечению 15-ти секунд.

Элементы для сборки схемы:

- Arduino UNO;
- беспаячная макетная плата;
- светодиоды: 2 зеленых, 1 желтый, 2 красных;
- 5 резисторов номиналом 220 Ом;
- 1 ультразвуковой датчик.

Критерии оценивания:

| № п/п | Критерий | Максимальный балл |
|----------|--|-------------------|
| 1 | Использование всех перечисленных в задании элементов. | 10 |
| 2 | Правильность сборки схемы устройства, аккуратность соединения элементов. | 10 |
| 3 | Приведение алгоритма работы программы в виде текста или блок-схемы. | 10 |

| | | |
|--------|--|----|
| 4 | Использование в коде программы директив define, условий, циклов, функций, комментариев для пояснения работы программы. | 10 |
| 5 | Приведение полного кода программы. Отсутствие ошибок в коде. Код структурирован, оформлен в едином стиле. | 20 |
| Итого: | | 60 |

Ответ на билет:

Схема собранного устройства:

На рисунке 1 приведена схема собранного устройства, состоящая из Arduino UNO, беспаячной макетной платы, светодиодов: 2 зеленых, 1 желтого, 2 красных; 5 резисторов номиналом 220 Ом; 1 ультразвукового дальномера.

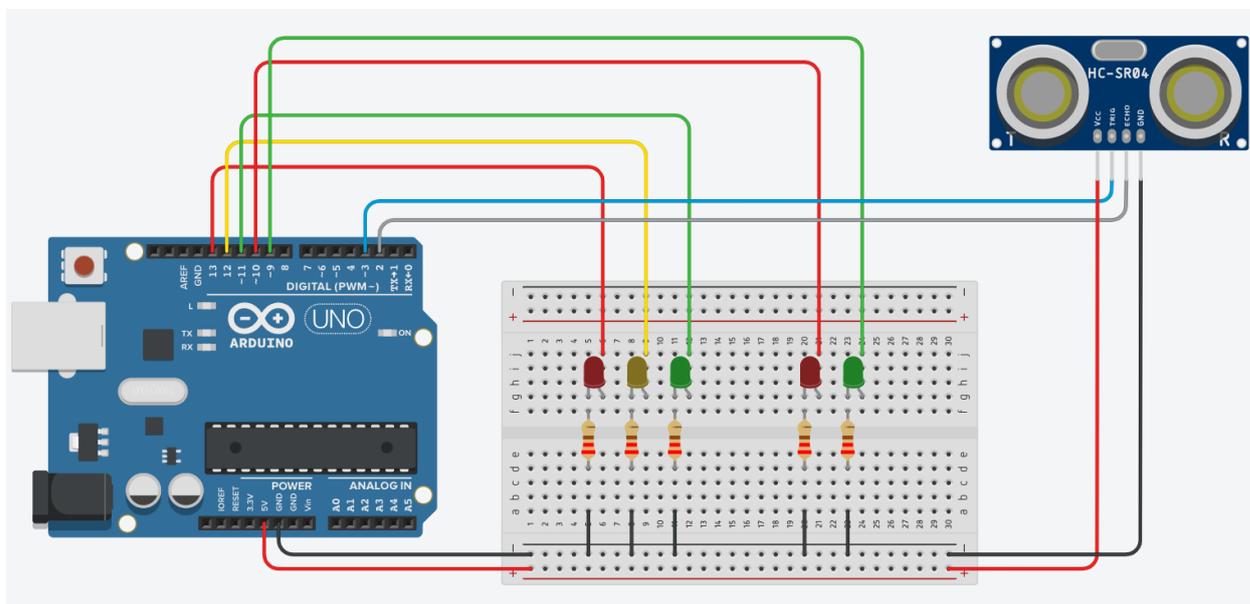


Рис. 1. Схема устройства.

Краткое описание подключенных пинов и основных функций:

Светодиоды подключены через резисторы 220 Ом к цифровым пинам 13, 12, 11, 10, 9. Контакты Echo и Trig ультразвукового дальномера подключены к цифровым пинам 2 и 3 соответственно. Для удобства введены директивы `auto_green`, `auto_yellow`, `auto_red`, `people_green`, `people_red`, `button`, которые задают номера соответствующих пинов. Приставка `auto` обозначает, что светодиоды предназначены для автомобилей; приставка `people` – для пешеходов; `green`, `yellow`, `red` – цвета светодиодов (светофора); `trigPin`, `echoPin` – пины ультразвукового дальномера.

Введена функция `blink`, содержащая цикл, отвечающий за мигание светодиода. В нее передается переменная типа `int`, соответствующая номеру пина с подключенным светодиодом.

В функции `setup` определяются режимы работы пинов. Для

светодиодов и Trig пина дальномера – `OUTPUT`, т.к. они являются средством вывода информации, для пина Echo – `INPUT`, поскольку этот пин является средством ввода информации.

В функции loop описывается алгоритм работы светофора. Изначально задаем зеленый свет для автомобилей и красный для пешеходов. Для работы устройства сравнения генерируется и принимается ультразвуковая волна дальномера, после чего рассчитывается расстояние до пешехода. Если расстояние до пешехода больше 1-го метра (выполнение условия if), то продолжают гореть красный светодиод для пешеходов и зеленый для автомобилей. Если пешеход приблизился на расстояние меньше 1-го метра (выполняется условие else), то сменяется сигнал светофора. Командой delay(15000) реализована задержка на 15 секунд, после чего идет переключение светофора на разрешающий свет для автомобилей, который будет гореть в течение 15-ти секунд. После чего возобновляется считывание расстояния до пешехода.

Алгоритм работы программы:



Рис. 2. Блок-схема алгоритма программы.

Полный код программы:

```
// Введение глобальных переменных для пинов светодиодов и кнопки

// Введение глобальных переменных для пинов светодиодов и ультразвукового датчика

# define auto_green 11

# define auto_yellow 12

# define auto_red 13

# define people_green 9

# define people_red 10

# define trigPin 3

# define echoPin 2

// Функция мигания для зеленых светодиодов

void blink(int blink_LED)

{

// Задаем цикл на три мигания

for (int i=0; i<3; i++)

{

digitalWrite(blink_LED, LOW); //выключение светодиода

delay(1000); // задержка на 1 секунду

digitalWrite(blink_LED, HIGH); //включение светодиода

delay(1000); // задержка на 1 секунду

}

}
```

```
void setup()

{

// Определение режимов работы выводов пинов для светодиодов ультразвукового дальномера

pinMode(auto_red, OUTPUT);

pinMode(auto_yellow, OUTPUT);

pinMode(auto_green, OUTPUT);

pinMode(people_red, OUTPUT);

pinMode(people_green, OUTPUT);

// Определение режимов работы пинов для ультразвукового дальномера

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

// Включение монитора последовательного интерфейса

Serial.begin (9600);

}

void loop()

{

    digitalWrite(people_red, HIGH); // Начальные условия работы светофора

    digitalWrite(auto_green, HIGH); // Начальные условия работы светофора

// Генерирование и приём ультразвуковой волны

int duration, cm; //объявление переменной типа int
```

```

digitalWrite(trigPin, LOW); //выключение пина trigPin

    delayMicroseconds(2); //приостановка выполнения программы на 2мкс

digitalWrite(trigPin, HIGH); //включение пина trigPin

    delayMicroseconds(10); // приостановка выполнения программы на 10мкс

digitalWrite(trigPin, LOW); //выключение пина trigPin

/*

Присвоение переменной duration значения, выдаваемого функцией pulseIn;

Функция pulseIn используется для считывания длительности импульса

сигнала заданного уровня на заданном пине

*/

    duration = pulseIn(echoPin, HIGH);

    cm = duration / 58; //перевод значения duration в сантиметры

if (cm > 100) // сравнение значения расстояния с пороговым значением
{

digitalWrite(people_red, HIGH); // действие, в случае выполнения условия

digitalWrite(auto_green, HIGH); // действие, в случае выполнения условия

}

else // действия, в случае невыполнения условия

{

//Смена светофора на разрешающий для пешеходов

blink(auto_green); //вызов функции для мигания зеленого светофора для автомобилей

digitalWrite(auto_green, LOW); //выключение светодиода

```

```
digitalWrite(auto_yellow,HIGH); //включение светодиода
```

```
delay(1000); //задержка на 1 секунду
```

```
digitalWrite(auto_yellow,LOW); //выключение светодиода
```

```
digitalWrite(auto_red,HIGH); //включение светодиода
```

```
delay(1000); //задержка на 1 секунду
```

```
digitalWrite(people_red,LOW); //выключение светодиода
```

```
digitalWrite(people_green,HIGH); //включение светодиода
```

```
delay(15000); //задержка на 15 секунд
```

```
// Смена светофора на разрешающий для автомобилей
```

```
blink(people_green); //вызов функции для мигания зеленого светофора для пешеходов
```

```
digitalWrite(people_green,LOW); //выключение светодиода
```

```
digitalWrite(people_red, HIGH); //включение светодиода
```

```
delay(1000); //задержка на 1 секунду
```

```
digitalWrite(auto_red,LOW); //выключение светодиода
```

```
digitalWrite(auto_green, HIGH); //включение светодиода
```

```
delay(15000); //задержка на 15 секунд для работы зеленого светофора автомобилей
```

```
}
```

```
}
```

Кейс №2. Курс “Основы программирования на C++”

Задача 1

Выписать из текста все слова, начинающиеся на указанную букву (без учета регистра).

Формат ввода

В первой строке в файле `input.txt` записана одна латинская буква (строчная или заглавная). Далее записаны строки текста на латинском языке.

Формат вывода

Вывести на экран или в файл `output.txt` через пробел все слова, начинающиеся на указанную букву. Регистр букв в словах должен быть сохранен как в оригинале.

Пример входных и выходных данных

| Ввод | Вывод |
|--|-------------------------------------|
| m My parents are very good and kind. They are young. My mother's name is Natalia. She is thirty years old. She is a doctor. My mother is very beautiful. My father's name is Victor. He is thirty-two years old. He is an engineer. I love my parents very much. | My My mother's My mother My my much |

Задача 2

Вычислите сумму нечетных элементов, расположенных по периметру прямоугольной таблицы размеров $M \times N$.

| | | | | | |
|----|----|----|----|----|----|
| 7 | 3 | 17 | 11 | 2 | 3 |
| 1 | 8 | 12 | 5 | 1 | 15 |
| 6 | 16 | 17 | 10 | 12 | 4 |
| 8 | 15 | 18 | 1 | 1 | 3 |
| 20 | 17 | 17 | 7 | 9 | 9 |

Желтым на изображении выделены элементы, расположенные по периметру таблицы.

Формат ввода

В первой строке вводится два целых числа (M и N) - количество строк и столбцов таблицы соответственно. ($0 < M < 20$, $0 < N < 20$)

В последующих M строках через пробел вводятся элементы очередной строки - натуральные числа, не превышающие 10 000.

Формат вывода

Вывести одно число - сумму нечетных элементов, расположенных по периметру таблицы.

Пример входных и выходных данных

| Ввод | Вывод |
|--|--------------|
| 4 5 90 29 5 3 76 59 82 41 42 13 36 2 49 53 12 65 58 72 73 46 | 247 |

Критерии оценивания:

| № | Компетенция | Максимальный балл |
|---------------|--|-------------------|
| 1 | Умение считывать данные из файла, обрабатывать строки. | 30 |
| 2 | Умение работать с двумерными массивами. | 30 |
| Итого: | | 60 |

Примечание:

Каждая из задач проверяется на 10 тестах. Каждый тест, включая тест, приведенный в условии в качестве примера, оценивается в 3 балла.

К каждой задаче прилагается набор файлов с тестами и ответами на тесты. Учащимся доступен только один набор тест + ответ.

Варианты решения:

Задача 1

```
#include <iostream>

#include <vector>

#include <map>

#include <fstream>

using namespace std;

int main()
{
    ifstream fin("input.txt");

    map <string, vector<int>> A;

    string s, sch, st="";

    char ch;

    getline(fin, sch);
```

```

ch =sch[0];

while (getline(fin, s)){

st = "";

for (char& el: s){

    if (el == ' ' || el== '\\0'){

        if (tolower(st[0]) == ch)

            cout << st << " ";

        st = "";

    }

    else{

        if (isalpha(el) or el == '\\')

            st += el;

    }

}

if (tolower(st[0]) == ch)

    cout << st << " ";

}

return 0;

}

```

Задача 2

```

#include <iostream>

using namespace std;

#include <math.h>

#include <iomanip>

```

```
#include <vector>

#include <string>

int main() {

    int M, N;

    cin >> M >> N;

    vector<vector<int>> A;

    A.resize(M);

    for (int i=0; i<M; i++){
    for (int j=0; j<N; j++){

        int a;

        cin >> a;

        A[i].push_back(a);

    }

    }

    int s = 0;

    if (M == 1 && N == 1){

    if (A[0][0] % 2 != 0) s = A[0][0];

    }

    else if(M == 1){

    for (int j=0; j<N; j++){

        if (A[0][j] % 2 != 0) s += A[0][j];

    }

    }

    else if(N == 1){
```

```
for (int i=0; i<M; i++){
    if (A[i][0] % 2 != 0) s += A[i][0];
}
}
else{
for (int j=0; j<N; j++){

    if (A[0][j] % 2 != 0) s+= A[0][j];
    if (A[M-1][j] % 2 != 0) s+= A[M-1][j];
}

for (int i=1; i<M-1; i++){

    if (A[i][0] % 2 != 0) s+= A[i][0];
    if (A[i][N-1] % 2 != 0) s+= A[i][N-1];
}
}

cout << s;
}
```

Кейс №3. Курс “Основы программирования на Python”

Разработать на языке Python программу с консольным интерфейсом для учета успеваемости группы школьников по трём дисциплинам. Информация об учениках представляет собой совокупность следующих значений:

| Номер | 0 | 1 | 2 | 3 | 4 |
|-------|-------------|--------------|-----------------------------|----------------------------|------------------------|
| Поле | ФИО ученика | Год рождения | Средний балл по информатике | Средний балл по математике | Средний балл по физике |

При написании программы каждая команда реализуется в виде отдельной функции, все функции пишутся в одном общем файле вместе с основной программой.

Программа должна позволять выполнять следующие команды:

1. ДОБАВИТЬ <ФамилияИО> <гр> <ср_балл_инф> <ср_балл_матем> <ср_балл_физ> – добавляет ученика с заданными значениями.
2. ПОКАЗАТЬ <ФамилияИО> – выводит в консоль всю информацию об ученике с заданными фамилией и инициалами.
3. УДАЛИТЬ <ФамилияИО> – удаляет ученика с заданными фамилией и инициалами.
4. СВОДКА <номер_поля_сорт> – выводит в консоль информацию обо всех добавленных учениках, сортируя строки по заданному полю по возрастанию. Если поле строковое, то сортировка должна выполняться по алфавиту.
5. СОХРАНИТЬ <имя_файла> – сохраняет добавленных учеников в заданный текстовый файл в формате команды ДОБАВИТЬ (с разделением значений пробелами): <ФамилияИО> <гр> <ср_балл_инф> <ср_балл_матем> <ср_балл_физ>.
6. ЗАГРУЗИТЬ <имя_файла> – загружает данные об учениках из заданного текстового файла, полностью замещая текущий состав записей об учениках. Считается, что загружаемый файл был создан при вызове команды СОХРАНИТЬ.

Описание параметров команд:

- <ФамилияИО> – фамилия и инициалы ученика БЕЗ разделения пробелами (поле №0), например, ПетровСА. Данное значение считается уникальным для каждого ученика.
- <гр> – год рождения ученика, натуральное четырёхзначное число (поле №1).
- <ср_балл_инф> – средний балл по информатике, вещественное число на отрезке [2; 5] с точностью до сотых (поле №2).
- <ср_балл_матем> – средний балл по математике, вещественное число на отрезке [2; 5] с точностью до сотых (поле №3).
- <ср_балл_физ> – средний балл по физике, вещественное число на отрезке [2; 5] с точностью до сотых (поле №4).
- <номер_поля_сорт> – номер поля, по которому необходимо провести сортировку, целое число с вариантами значения от 0 до 4.
- <имя_файла> – путь (абсолютный или относительный) до текстового файла.

Предполагается, что программе на вход подаются только корректные (не вызывающие ошибок) последовательности команд.

Пример выполнения программы

| Ввод | Вывод |
|------|-------|
|------|-------|

| | |
|---|---|
| ДОБАВИТЬ ПетровСА 2005 2.71 3.14 4 ДОБАВИТЬ ИвановПС 2006 4.5 3.9 5 ДОБАВИТЬ ОшибкаВВ 2004 2 3 4 СВОДКА 0 УДАЛИТЬ ОшибкаВВ ПОКАЗАТЬ ПетровСА СОХРАНИТЬ db.txt УДАЛИТЬ ПетровСА СВОДКА 1 ЗАГРУЗИТЬ db.txt СВОДКА 2 | ИвановПС 2006 4.5 3.9 5.0 ОшибкаВВ 2004 2.0 3.0 4.0 ПетровСА 2005 2.71 3.14 4.0 ПетровСА 2005 2.71 3.14 4.0 ИвановПС 2006 4.5 3.9 5.0 ПетровСА 2005 2.71 3.14 4.0 ИвановПС 2006 4.5 3.9 5.0 |
|---|---|

Критерии оценки:

Реализация каждой из шести команды оценивается отдельно по следующему принципу:

- команда реализована как функция и выполняется корректно – 10 баллов;
- команда реализована не как функция и выполняется корректно – 5 баллов;
- команда выполняется не корректно при любом способе реализации – 0 баллов.

| № п/п | Команда | Реализована как функция и выполняется корректно | Реализована не как функция и выполняется корректно | Выполняется не корректно при любом способе реализации |
|--------------|-----------|---|--|---|
| 1 | ДОБАВИТЬ | 10 баллов | 5 баллов | 0 баллов |
| 2 | ПОКАЗАТЬ | 10 баллов | 5 баллов | 0 баллов |
| 3 | УДАЛИТЬ | 10 баллов | 5 баллов | 0 баллов |
| 4 | СВОДКА | 10 баллов | 5 баллов | 0 баллов |
| 5 | СОХРАНИТЬ | 10 баллов | 5 баллов | 0 баллов |
| 6 | ЗАГРУЗИТЬ | 10 баллов | 5 баллов | 0 баллов |
| ИТОГО | | 60 баллов | 30 баллов | 0 баллов |

Кейс №4. Курс “Язык программирования Java”

«Программа для хранения контактных данных»

Разработать на языке Java программу с консольным интерфейсом для хранения контактных данных физических лиц. Контактные данные представляют собой совокупность следующих значений:

| Номер | 0 | 1 | 2 | 3 | 4 |
|-------|--------------|---------|--------|------------------|--------------|
| Поле | ФИО человека | Группа* | E-mail | Телефонный номер | Год рождения |

* под группой понимается строковое значение для классификации контакта: «родственник», «коллега», «друг» и т.п.

При написании программы каждая команда реализуется в виде отдельного метода, все методы в единственном классе, содержащем метод main. Программа должна позволять выполнять следующие команды:

1. INSERT <ФамилияИО> <группа> <email> <тел> <гр> – добавляет данные о новом контакте с заданными значениями.
2. ECHO <ФамилияИО> – выводит в консоль всю информацию о контакте с заданными фамилией и инициалами.
3. DEL <ФамилияИО> – удаляет контакт с заданными фамилией и инициалами.
4. SORT_PRINT <номер_поля_сорт> – выводит в консоль информацию обо всех добавленных контактах, сортируя строки по заданному полю по возрастанию. Если поле строковое, то сортировка должна выполняться по алфавиту.
5. INTO_FILE <имя_файла> – сохраняет добавленные контакты в заданный текстовый файл в формате команды INSERT (с разделением значений пробелами): <ФамилияИО> <группа> <email> <тел> <гр>.
6. FROM_FILE <имя_файла> – загружает контактные данные из заданного текстового файла, полностью замещая текущий состав записей о контактах. Считается, что загружаемый файл был создан при вызове команды INTO_FILE.

Описание параметров команд:

- <ФамилияИО> – фамилия и инициалы контакта БЕЗ разделения пробелами (поле №0), например, ПетровСА. Данное значение является уникальным для каждого контакта.
- <группа> – классификация контакта, строковое значение, не содержащее пробелов (поле №1).
- <email> – адрес электронной почты, строковое значение, не содержащее пробелов (поле №2). Проверять корректность формата не требуется.
- <тел> – телефонный номер контакта, строковое значение, не содержащее пробелов (поле №3). Проверять корректность формата не требуется.
- <гр> – год рождения физического лица (контакта), натуральное четырёхзначное число (поле №4).
- <номер_поля_сорт> – номер поля, по которому необходимо провести сортировку, целое число с вариантами значения от 0 до 4.
- <имя_файла> – путь (абсолютный или относительный) до текстового файла.

Предполагается, что программе на вход подаются только корректные (не вызывающие ошибок) последовательности команд.

Пример выполнения программы

| Ввод | Вывод |
|---|--|
| INSERT ПетровСА друг pet@rov.ru 1(234)567-89-10 1997 INSERT ИвановПС друг iv@nov.ru 2(134)567-89-10 1996 INSERT ОшибкаВВ коллега 404@error.ru 0(000)000-00-00 1970 SORT_PRINT 0 DEL ОшибкаВВ ECHO ПетровСА INTO_FILE db.txt DEL ПетровСА SORT_PRINT 1 FROM_FILE db.txt SORT_PRINT 2 | ИвановПС друг iv@nov.ru 2(134)567-89-10 1996 ОшибкаВВ коллега 404@error.ru 0(000)000-00-00 1970 ПетровСА друг pet@rov.ru 1(234)567-89-10 1997 ПетровСА друг pet@rov.ru 1(234)567-89-10 1997 ИвановПС друг iv@nov.ru 2(134)567-89-10 1996 ИвановПС друг iv@nov.ru 2(134)567-89-10 1996 ПетровСА друг pet@rov.ru 1(234)567-89-10 1997 |

Критерии оценки:

Реализация каждой из шести команд оценивается отдельно по следующему принципу:

- команда реализована как отдельный метод и выполняется корректно – 10 баллов;
- команда реализована не как отдельный метод и выполняется корректно – 5 баллов;
- команда выполняется не корректно при любом способе реализации – 0 баллов.

| № п/п | Команда | Реализована как метод и выполняется корректно | Реализована не как метод и выполняется корректно | Выполняется не корректно при любом способе реализации |
|--------------|------------|---|--|---|
| 1 | INSERT | 10 баллов | 5 баллов | 0 баллов |
| 2 | ECHO | 10 баллов | 5 баллов | 0 баллов |
| 3 | DEL | 10 баллов | 5 баллов | 0 баллов |
| 4 | SORT_PRINT | 10 баллов | 5 баллов | 0 баллов |
| 5 | INTO_FILE | 10 баллов | 5 баллов | 0 баллов |
| 6 | FROM_FILE | 10 баллов | 5 баллов | 0 баллов |
| ИТОГО | | 60 баллов | 30 баллов | 0 баллов |