

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)

Институт информатики и вычислительной техники

ЗАМЕНИТЬ 09.03.01 "Информатика и вычислительная  
техника"  
профиль "Программное обеспечение мобильных систем "

**РГЗ**

по дисциплине «Информатика»

Выполнил:

Студент гр. ИКС-432

\_\_\_\_\_/Салий В. П./  
ФИО студента

«\_\_» \_\_\_\_\_ 2024 г.

Проверил:

Старший преподаватель  
кафедры ПМиК

\_\_\_\_\_/Агалаков А. А./  
ФИО преподавателя

«\_\_» \_\_\_\_\_ 2024 г.

Оценка \_\_\_\_\_

Новосибирск 2024

## Оглавление

Задание	3
Теория	<b>Ошибка! Закладка не определена.</b>
Ход работы	5
Демонстрация работы	<b>Ошибка! Закладка не определена.</b>
Вывод	9
Литература	9
Приложение	11

## Задание

3. Футбольный турнир Вам необходимо написать программу, в которой будут реализованы следующие функции: а. При нажатии на кнопку “Добавить команду” появляется диалоговое окно, в которое пользователь вводит название новой команды. (Сделать проверку, что такой команды в списке еще нет) б. При нажатии на кнопку “Удалить команду” появляется диалоговое окно, в которое пользователь вводит название команды, которую хочет удалить из списка. в. При нажатии на кнопку “Экспорт в файл” список команд сохраняется в файл, который указал пользователь. г. При нажатии на кнопку “Импорт из файла” происходит заполнение списка команд из файла, который указал пользователь. При этом должно появиться диалоговое окно с вопросом “действительно ли вы хотите загрузить список команд из файла? Все команды, которые были вами добавлены и не сохранены в файл будут удалены”. Список команд в программе очищается и заполняется командами из файла. д. При нажатии на кнопку “Создать турнир”, появляется диалоговое окно, в котором пользователь может указать название турнира, и количество команд, участвующих в нем. После создается файл, именем которого является название турнира, со следующим содержанием: НАЗВАНИЕ ТУРНИРА Команда 1 - Команда 2 Команда 3 - Команда 4 Команда 5 - Команда 1 \*Название команд берётся рандомным образом из списка.

## Теория

Название рисунка дается под рисунком. Пожалуйста, не включайте название в состав самого рисунка, а также не используйте рамок вокруг рисунков.

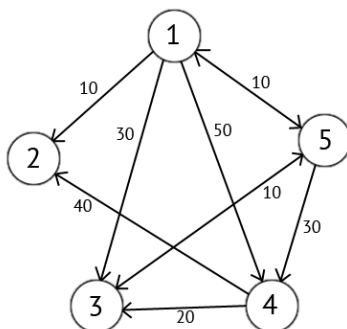


Рисунок 1. – Ориентированный граф G

Заголовок таблицы дается над таблицей. Таблицы и рисунки располагаются по центру колонки.

Таблица 1. Назначение нитей серверного процесса

Нить	Назначение	Приор.	Актив	Статус
1	Прием запросов	30	-	
2 – 10	Обслуживание запросов	10	sem_1	ок

Названия полей таблицы набираются шрифтом 11 пт. При ссылках на рисунки и таблицы используйте сокращения «рис.» и «табл.».

## Ход работы программы

### 1. Запуск программы:

При запуске создается экземпляр класса `TournamentApp`, который инициализирует интерфейс.

```
if __name__ == "__main__":  
    app = TournamentApp()  
    app.run()
```

### 2. Создание интерфейса:

В методе `create_layout` создаются кнопки и список команд.

```
def create_layout(self):  
    layout = [  
        [sg.Button("Добавить команду"), sg.Button("Удалить команду")],  
        [sg.Button("Экспорт в файл"), sg.Button("Импорт из файла")],  
        [sg.Button("Создать турнир"), sg.Button("Эмуляция турнира")],  
        [sg.Listbox(values=self.teams, size=(30, 10), key="-TEAM_LIST-", enable_events=True)],  
        [sg.Button("Выход")]  
    ]  
    return layout
```

### 3. Добавление команды:

При нажатии кнопки «Добавить команду» вызывается метод `add_team`, который открывает диалог для ввода названия команды.

```
def add_team(self):  
    team_name = sg.popup_get_text("Введите название команды:")  
    if team_name and team_name not in self.teams:  
        self.teams.append(team_name)  
        self.update_team_list()  
    else:  
        sg.popup_error("Ошибка: такая команда уже есть или название пустое.")
```

#### 4. Удаление команды

Пользователь выбирает команду из списка и нажимает кнопку «Удалить команду». Вызывается метод `remove_team`.

```
def remove_team(self):
    selected_team = self.window["-TEAM_LIST-"].get()[0] if self.window["-TEAM_LIST-"].get() else None
    if selected_team:
        self.teams.remove(selected_team)
        self.update_team_list()
    else:
        sg.popup_error("Ошибка: выберите команду для удаления.")
```

#### 5. Экспорт в файл:

При нажатии кнопки «Экспорт в файл» вызывается метод `export_to_file`

```
def export_to_file(self):
    file_name = sg.popup_get_file("Сохранить как", save_as=True, no_window=True)
    if file_name:
        with open(file_name, 'w') as f:
            for team in self.teams:
                f.write(team + '\n')
        sg.popup("Команды экспортированы в файл.")
```

#### 6. Импорт из файла:

При нажатии кнопки «Импорт из файла» вызывается метод `import_from_file`

```
def import_from_file(self):
    file_name = sg.popup_get_file("Выберите файл для импорта", no_window=True)
    if file_name:
        confirm = sg.popup_yes_no("Вы уверены, что хотите загрузить список команд из файла? Все несохраненные команды будут удалены.")
        if confirm == "Yes":
            with open(file_name, 'r') as f:
                self.teams = [line.strip() for line in f.readlines() if line.strip()]
            self.update_team_list()
```

## 7. Создание турнира:

При нажатии кнопки «Создать турнир» вызывается метод `create_tournament`.

```
def create_tournament(self):
    tournament_name = sg.popup_get_text("Введите название турнира:")
    num_teams = sg.popup_get_text("Введите количество команд:")

    try:
        num_teams = int(num_teams)
        if num_teams > len(self.teams) or num_teams < 2 or num_teams % 2 != 0:
            sg.popup_error("Ошибка: должно быть четное количество команд, больше одной.")
            return

        matches = []
        random_teams = random.sample(self.teams, num_teams)
        for i in range(0, num_teams, 2):
            if i + 1 < num_teams:
                matches.append(f"{random_teams[i]} - {random_teams[i + 1]}")

        with open(f"{tournament_name}.txt", 'w') as f:
            f.write(f"{tournament_name}\n")
            for match in matches:
                f.write(match + '\n')
        sg.popup(f"Турнир '{tournament_name}' создан.")
    except ValueError:
        sg.popup_error("Ошибка: введите корректное число.")
```

## 8. Эмуляция турнира:

При нажатии кнопки «Симуляция турнира» вызывается метод `simulate_tournament`.

```
def simulate_tournament(self):
    tournament_name = sg.popup_get_text("Введите название турнира:")
    num_teams = sg.popup_get_text("Введите количество команд:")

    try:
        num_teams = int(num_teams)
        if num_teams > len(self.teams) or num_teams < 2 or num_teams % 2 != 0:
            sg.popup_error("Ошибка: должно быть четное количество команд, больше одной.")
            return

        matches = []
        scores = {}
        random_teams = random.sample(self.teams, num_teams)
        for i in range(0, num_teams, 2):
            if i + 1 < num_teams:
                score1 = random.randint(0, 5)
                score2 = random.randint(0, 5)
                matches.append(f"{random_teams[i]} - {random_teams[i + 1]} Счет: {score1} - {score2}")
                scores[random_teams[i]] = scores.get(random_teams[i], 0) + score1
                scores[random_teams[i + 1]] = scores.get(random_teams[i + 1], 0) + score2

        with open(f"{tournament_name}.txt", 'w') as f:
            f.write(f"{tournament_name}\n")
            for match in matches:
                f.write(match + '\n')

        max_goals = max(scores.values(), default=0)
        winners = [team for team, score in scores.items() if score == max_goals]
        f.write("Победители: " + ', '.join(winners) + '\n')
        sg.popup(f"Эмуляция турнира '{tournament_name}' завершена.")
    except ValueError:
        sg.popup_error("Ошибка: введите корректное число.")
```

## 9. Обновление списка команд:

Метод `update_team_list` обновляет отображаемый список команд в интерфейсе

```
def update_team_list(self):
    self.window["-TEAM_LIST-"].update(self.teams)
```

## Демонстрация работы

Скрины работы программы с пояснениями.



## Вывод

Вывод о проделанной работе.

## Литература

Список литературы оформляется в соответствии с ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила составления». Описание зарубежных источников дается на языке оригинала без перевода. При этом в названии статьи заглавной пишется только первая буква (за исключением имен собственных и символов элементов). Для описаний работ в переводимых журналах указывайте сначала русскоязычный вариант, а затем (при необходимости) вариант на иностранном языке. Адреса в Интернете рекомендуется вводить моноширинным шрифтом Courier New 11 пт или аналогичным.

1. Рябко Б. Я., Фионов А. Н. Криптографические методы защиты информации: учебное пособие для вузов. М.: Горячая линия–Телеком, 2005. 229 с.
2. Blake I., Seroussi G., Smart N. Elliptic Curves in Cryptography. Cambridge University Press, 2002. 204 p.
3. История России: учебник / А. С. Орлов, В. А. Георгиев, Н. Г. Георгиева, Т. А. Сивохина. 2-е изд., перераб. и доп. М.: Проспект, 2004. 514 с.
4. Экономика: учебник / под ред. А. С. Булатова. 3-е изд., перераб. и доп. М.: Экономистъ, 2003. 894 с.
5. Рябко Б. Я., Фионов А. Н. Эффективный метод адаптивного арифметического кодирования для источников с большими алфавитами // Проблемы передачи информации. 1999. Т. 35, № 4. С. 1–14.
6. Лэтчфорд Е. У. С Белой армией в Сибири [Электронный ресурс] // Восточный фронт армии адмирала А. В. Колчака: [сайт]. [2004]. URL:

<http://east-front.narod.ru/memo/latchford.htm>  
обращения: 23.08.2007).

(дата

## Приложение

**Обязательно с новой страницы!** Код программы с комментариями, а также code [style!](#) Использовать Courier New шрифт 12.

```
#include <stdio.h>
int main()
{
    printf("Здравствуй, мир!");
    getchar();
    return 0;
}
```