

Міністерство освіти і науки України
Національний технічний університет України “Київський політехнічний
інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №5

з дисципліни

“Програмування інтелектуальних інформаційних систем”

Виконала

ІІ-21 Скрипець О. О.

Перевірив

Баришич Л. М.

Київ 2024

Завдання

- 1 Зіскрапити заголовки новин з сайту (<https://pestrecy-rt.ru/news/tag/list/specoperaciia/>) <https://archive.is/pestrecy-rt.ru> за допомогою xpath (можливо знадобиться vpn)
- 2 Реалізувати кроулінг через натискання на кнопку “Далі”
- 3 По отриманих заголовках створити список померлих росіян

Скрапити і кроулити можна як створивши застосунок, так і використовуючи плагіни чи сторонні ресурси

Виконання

1. Зберігаю у файл всі заголовки з сайту, проходячись по кожній сторінці

```
const axios = require("axios");
const cheerio = require("cheerio");
const fs = require("fs");

async function getNewsHeadlinesFromPage(page) {
  try {

    const url = `https://pestrecy-rt.ru/news/tag/list/specoperaciia/page/${page}`;
    const { data } = await axios.get(url);
    const $ = cheerio.load(data);

    const headlines = [];
    $(".all-news__list_text-container h2").each((index, element) => {
      const headline = $(element).text().trim();
      headlines.push(headline);
    });

    return headlines;
  } catch (error) {
    console.error(`Помилка при отриманні новин з сторінки ${page}:`, error);
  }
}
```

```

        return [];
    }
}

async function getAllNewsHeadlines() {
    const allHeadlines = [];

    for (let page = 1; page <= 19; page++) {
        console.log(`Отримуємо заголовки зі сторінки ${page}...`);
        const headlines = await getNewsHeadlinesFromPage(page);
        allHeadlines.push(...headlines);
    }

    const filePath = "headlines.txt";
    fs.writeFile(filePath, allHeadlines.join("\n"), (err) => {
        if (err) {
            console.error("Помилка при записі у файл:", err);
        } else {
            console.log(`Заголовки успішно записані у файл:
${filePath}`);
        }
    });
}

getAllNewsHeadlines();

```

2. Зберігаю у файл всі підзаголовки з сайту, проходячись по кожній сторінці

```

const axios = require("axios");
const cheerio = require("cheerio");
const fs = require("fs");

async function getNewsSubheadlinesFromPage(page) {
    try {
        const url = `https://pestrecy-
rt.ru/news/tag/list/specoperaciia/page/${page}`;
        const { data } = await axios.get(url);
    }
}

```

```

    const $ = cheerio.load(data);
    const subheadlines = [];
    $(".all-news__list_text-container
p.oneNews__link").each(
        (index, element) => {
            const subheadline = $(element).text().trim();
            subheadlines.push(subheadline);
        }
    );

    return subheadlines;
} catch (error) {
    console.error(
        `Помилка при отриманні підзаголовків з сторінки
${page}:`,
        error
    );
    return [];
}
}

async function getAllNewsSubheadlines() {
    const allSubheadlines = [];

    for (let page = 1; page <= 19; page++) {
        console.log(`Отримуємо підзаголовки зі сторінки
${page}...`);
        const subheadlines = await
getNewsSubheadlinesFromPage(page);
        allSubheadlines.push(...subheadlines);
    }

    const filePath = "subheadlines.txt";
    fs.writeFile(filePath, allSubheadlines.join("\n"), (err)
=> {
        if (err) {
            console.error("Помилка при записі у файл:", err);
        } else {
            console.log(`Підзаголовки успішно записані у файл:
${filePath}`);

```

```
    }  
  });  
}  
  
getAllNewsSubheadlines();
```

3. Відбираю імена та прізвища за допомогою ключових слів

```
const fs = require("fs");  
  
const headlines = fs.readFileSync("headlines.txt", "utf-8").split("\n");  
const subheadlines = fs.readFileSync("subheadlines.txt", "utf-8").split("\n");  
  
const namePattern = /([А-ЯЁ][а-яё]+\s+[А-ЯЁ][а-яё]+)/g;  
  
const deathKeywords = [  
  "загибель",  
  "пам'ять",  
  "скорбь",  
  "печаль",  
  "трагедия",  
  "утрата",  
  "панихида",  
  "скорбящие",  
  "скорбящий",  
  "несчастье",  
  "несчастный случай",  
  "отдавая дань",  
  "соболезнования",  
  "сердечное соболезнование",  
  "слезы",  
  "траурная церемония",  
  "поминальная служба",  
  "покойный",  
  "покойная",  
  "вечная память",  
  "траурная лента",
```

```
"памятник",
"лежит под землей",
"грустная новость",
"вечный покой",
"похоронная процессия",
"прощание с покойным",
"ушёл из жизни",
"погиб",
"смерть",
"простились",
"погибшим",
"умер",
"убит",
"скончался",
"похороны",
"захоронение",
"погибшие",
"траур",
"проводили в последний путь",
"жертва",
"гибель",
"прощание",
"прощались",
"пал",
"посмертно",
"вдова",
"вдовец",
"погибшего",
"павшие",
"Светлая память",
];

const names = new Set();

const findNames = (textArray) => {
  textArray.forEach((line) => {

    const containsDeathKeyword = deathKeywords.some((keyword)
=>
      line.toLowerCase().includes(keyword)
```

```

    });
    if (containsDeathKeyword) {
        const matches = line.match(namePattern);
        if (matches) {
            matches.forEach((name) => names.add(name));
        }
    }
    });
};

findNames(headlines);
findNames(subheadlines);

console.log("Знайдені імена та прізвища загиблих:");
names.forEach((name) => console.log(name));

```

Результат

```

[Running] node "c:\Users\olya\Desktop\good
javascript\findname.js"
Знайдені імена та прізвища загиблих:
Александром Агафоновым
Валерием Межва
Собир Худайбердыев
Юрия Ивановича
Марии Ивановны
Сергей Корчагин
Артема Прокопчука
Александр Владимиров

[Done] exited with code=0 in 0.259 seconds

```