

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
СӘТБАЕВ УНИВЕРСИТЕТІ

Автоматика және Информациондық Технологиялар институты
Программалық Инженерия кафедрасы



SATBAYEV
UNIVERSITY

ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС #4

Тақырыбы: Жетілдірілген рекурсия және функционалдық үлгілер

№	Жұмысты орындау сапасы	Баға диапазоны	Орындалған %
1	Орындалған жоқ	0%	
2	Орындалды	0-50%	
3	Материялдық өзіндік жүйелендіру	0-10%	
4	Талап етілген көлемде және көрсетілген мерзімде орындау	0-5%	
5	Қосымша ғылыми әдебиеттерді пайдалану	0-5%	
6	Орындаған тапсырманың ерекшелігі	0-10%	
7	СӨЖ-ді қорғау	0-20%	
	Қорытынды:	0-100%	

Оқытушы: Шаяхметов Д
Студент: Ұлдақан А
Мамандығы: Computer Science
Тобы: Дс 7:50 – 9:45

Алматы 2024 ж

Мақсат:

Python тіліндегі озық рекурсивті әдістер мен функционалдық үлгілерді терең меңгеру және қолдану. Зертхананың мақсаты рекурсивті кодты жазу дағдыларын жақсарту ғана емес, сонымен қатар бағдарламалардың оқылуын, тиімділігін және масштабталуын жақсартатын әртүрлі функционалдық үлгілерді түсіну және қолдану болып табылады.

Тапсырмалар:

1. Рекурсияны терең зерттеу:

- Күрделі рекурсивті есептер мен алгоритмдерді талдау.
- Python тілінде құйрықты рекурсия және оны жүзеге асыру принциптерін зерттеу.
- мәліметтерді өңдеу және алгоритмдік есептерді шешу үшін рекурсияны практикалық қолдану.

2. Функционалдық үлгілерді меңгеру:

- Монадалар, карринг және функционалдық композиция сияқты функционалдық үлгілерді зерттеу және қолдану.
- Функционалдық үлгілердің кодты ұйымдастыруға және құрылымына қалай әсер ететіні туралы түсінікті дамыту.

3. Функционалдық программалауды нақты есептерде қолдану:

- Практикалық есептерге күрделі функционалды шешімдерді енгізу, мысалы, деректерді талдау, мәтінді өңдеу немесе веб-қызметтерді құру.
- Модульділік пен кодтың сыналуды жақсарту үшін функционалдық тәсілдерді қолдану.

4. Жақсартылған код өнімділігі және оқылу мүмкіндігі:

- Рекурсия мен функционалдық заңдылықтардың бағдарламаның жұмысына әсерін талдау.
- Функционалдық бағдарламалау стилінің артықшылықтары мен кемшіліктерін кодты оқу және техникалық қызмет көрсету тұрғысынан бағалау.

5. Сыни тұрғыдан ойлау және проблемалық талдау дағдыларын дамыту:

- Күрделі есептерді шешу кезінде сыни талдау мен тәуелсіз ізденістерді ынталандыру.
- Теориялық білімдерін практикада қолдану және бейімдеу қабілетін дамыту.

Зертханалық жұмыстың маңыздылығы:

Зертханалық жұмыс функционалдық бағдарламалау мен рекурсияда алдыңғы қатарлы дағдыларды дамытуға бағытталған. Бұл студенттерге кодының сапасы мен ауқымдылығын жақсартып отырып, функционалды бағдарламалау принциптерін нақты әлемдегі мәселелерге қалай қолдануға болатынын жақсы түсінуге көмектеседі. Сондай-ақ жұмыс аналитикалық ойлауды дамытуға және әртүрлі бағдарламалық тапсырмалар мен талаптарға бейімделу қабілетіне ықпал етеді.

Жеке тапсырмалар:

Әрбір студентке топ тізіміндегі санына сәйкес бірегей тапсырма беріледі (SSO қараңыз). Студенттерге рекурсияда қосымша дағдыларды дамытуға және функционалдық заңдылықтарды түсінуге көмектесетін есептер:

2	2021-2022	Бак	6B06102 Computer Science	Ұлдақан Ален Серужанұлы	Полная
---	-----------	-----	--------------------------	-------------------------	--------

2. Сөмке мәселесін шешуге арналған рекурсиямен динамикалық бағдарламалау
- Сөмке мәселесін шешу үшін есте сақтау арқылы рекурсияны пайдаланыңыз.

```
def knapsack(weights, values, capacity):
    memo = {}

    def knapsack_helper(index, current_capacity):

        if index < 0 or current_capacity == 0:
            return 0

        if (index, current_capacity) in memo:
            return memo[(index, current_capacity)]

        if weights[index] > current_capacity:
            memo[(index, current_capacity)] = knapsack_helper(index - 1,
current_capacity)
            return memo[(index, current_capacity)]

        included = values[index] + knapsack_helper(index - 1, current_capacity -
weights[index])
        not_included = knapsack_helper(index - 1, current_capacity)

        memo[(index, current_capacity)] = max(included, not_included)
        return memo[(index, current_capacity)]

    return knapsack_helper(len(weights) - 1, capacity)

weights = [5, 4, 6, 3]
values = [10, 40, 30, 50]
capacity = 10
print(knapsack(weights, values, capacity))
```

```
PS C:\Users\Admin\OneDrive\Документы\6th semester\Functional Programming\4 Lab> python -u "c:\Users\Admin\OneDrive\Документы\6th semester\Functional Programming\4 Lab\knapsack.py"
90
PS C:\Users\Admin\OneDrive\Документы\6th semester\Functional Programming\4 Lab> █
```

\$ git init

```
Admin@Sam MINGW64 ~/OneDrive/Документы/6th semester/Functional Programming/4 Lab
$ git init
Initialized empty Git repository in C:/Users/Admin/OneDrive/Документы/6th semester/Funct
ional Programming/4 Lab/.git/
```

\$ git remote add origin <URL>

```
Admin@Sam MINGW64 ~/OneDrive/Документы/6th semester/Functional Programming/4 Lab (master
)
$ git remote add origin https://github.com/JackOptimist/Functional-Programming.git
```

\$ git checkout -b branch name (lab4)

```
Admin@Sam MINGW64 ~/OneDrive/Документы/6th semester/Functional Programming/4 Lab (master
)
$ git checkout -b lab4
Switched to a new branch 'lab4'
```

\$ git add knapsack.py

```
Admin@Sam MINGW64 ~/OneDrive/Документы/6th semester/Functional Programming/4 Lab (lab4)
$ git add knapsack.py
```

\$ git status

```
Admin@Sam MINGW64 ~/OneDrive/Документы/6th semester/Functional Programming/4 Lab (lab4)
$ git status
On branch lab4

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   knapsack.py
```

\$ git commit -m "<COMMIT>"

```
$ git commit -m "Just python file added"
[lab4 (root-commit) 0708ce0] Just python file added
1 file changed, 28 insertions(+)
create mode 100644 knapsack.py
```

\$ git push origin lab4 (branch name)

```
$ git push origin lab4
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 501 bytes | 501.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'lab4' on GitHub by visiting:
remote:   https://github.com/JackOptimist/Functional-Programming/pull/new/lab4
remote:
To https://github.com/JackOptimist/Functional-Programming.git
* [new branch]      lab4 -> lab4
```

The screenshot shows the GitHub interface for a repository named 'lab4'. At the top, there's a navigation bar with 'lab4', '5 Branches', and '0 Tags'. A search bar and 'Add file' and 'Code' buttons are also present. Below this, a status bar indicates 'This branch is 1 commit ahead of, 1 commit behind master'. The main content area shows a commit by 'JackOptimist' with the message 'Just python file added' and hash '0708ce0' from '2 minutes ago'. A file named 'knapsack.py' is highlighted with a red box. Below the commit, there's a 'README' section with a book icon.