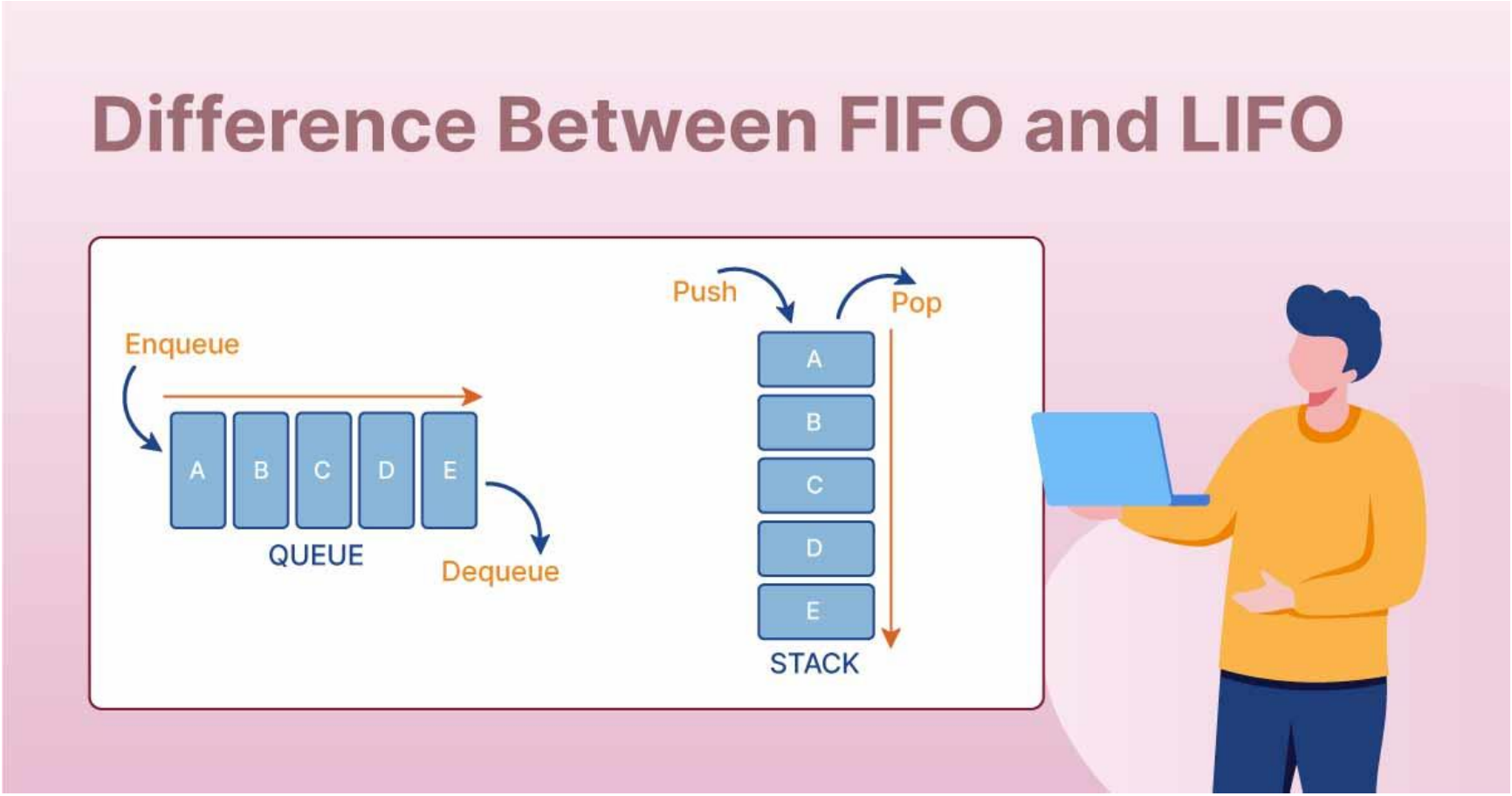
Duomenų struktūrų naudojimas.

Praėjusios pamokos santrauka

- FIFO, LIFO sąvokos
- stekai / deklai (dekas)
- jų skirtumai
- pagrindinės operacijos su steku / deku
- Python ir C++ implementacijų skirtumai

FIFO (First In, First Out)
LIFO (Last In, First Out)



Šaltinis: https://www.shiksha.com/online-courses/articles/difference-between-fifo-and-lifo/

Duomenų struktūros

- Suprasti pagrindines duomenų struktūrų sąvokas
  - masyvai
  - susietieji sąrašai
  - stekai / deklai (dekas)
  - eilės
- Įgyvendinimas naudojant Python ir C++
- Suprasti įvairių duomenų struktūrų privalumus ir trūkumus

Ką išmoksime šiandien

- Tyrinėsime, kaip veikia eilės
- Lyginsime stekų, deklų ir eilių funkcionalumą
- Darysime praktines užduotis, kad geriau suprastumėte jų skirtumus ir kaip juos pritaikyti sprendžiant įvairias problemas

Kas yra eilė?

- Eilė – tai **pirmo įėjimo, pirmo išėjimo** (FIFO) duomenų struktūra.
- Elementai pridedami į galą ir pašalinami iš priekio

Panašu į laukimo eilę, kur pirmas žmogus eilėje yra aptarnaujamas pirmas.

Dažnos operacijos

- **enqueue**: Pridėjimas į eilės galą.
- **dequeue**: Pašalinimas iš eilės priekio.
- **peek**: Peržiūrėti pirmąjį elementą eilėje, jo neištrinant.

Python ir C++ įgyvendinimas

- Python: Naudojame deque modulį eilės įgyvendinimui.
- C++: Naudojame std::queue klasę pagrindinėms eilės operacijoms.

Python – Eilės naudojant „deque“

```python
from collections import deque

# Sukuriame eilę
queue = deque()

# Pridedame elementus (enqueue)
queue.append("Element 1")
queue.append("Element 2")

# Pašaliname elementą (dequeue)
first = queue.popleft()

# Peržiūrime pirmą elementą
peek_element = queue[0]

print(f"Pirmas elementas: {peek_element}")
```

C++ – Eilės naudojant „std::queue“

```cpp
#include <iostream>
#include <queue>

int main() {
    std::queue<std::string> queue;

    // Pridedame elementus (enqueue)
    queue.push("Element 1");
    queue.push("Element 2");

    // Pašaliname elementą (dequeue)
    queue.pop();

    // Peržiūrime pirmą elementą
    std::cout << "Pirmas elementas: " << queue.front() << std::endl;

    return 0;
}
```

Svarbiausios operacijos

- **Python**: append() ir popleft().
- **C++**: push() ir pop().

Praktika

- Simuliuokite banko operacijų valdymo sistemą, naudodami eilę, steką ir deklą, kad pamatytumėte jų skirtumus atliekant tą pačią užduotį.
- **Scenarijus:**
  - Bankas turi apdoroti klientų transakcijas. Kiekvienas klientas gali atvykti, atlikti transakciją, arba atšaukti ją. Klientai turi būti aptarnaujami skirtingais prioritetais, priklausomai nuo pasirinktos duomenų struktūros

Scenarijus

Eilė:

- Klientai aptarnaujami atvykimo tvarka (FIFO).
- Naudojamas tais atvejais, kai klientai aptarnaujami pagal atvykimo laiką, kaip banko kasose.

Stekas:

- Paskutinis atvykęs klientas aptarnaujamas pirmas (LIFO).
- Naudojamas situacijose, kai prioritetas skiriamas paskutinei įvykusiai transakcijai, pvz., kai skubūs pavedimai yra svarbesni.

Deklas:

- Klientai gali būti aptarnaujami ir iš priekio, ir iš galo, atsižvelgiant į poreikius.
- Naudojamas, kai kartais pirmi klientai turi prioritetą, o kartais reikia aptarnauti paskutinius, pvz., VIP klientai gali būti praleidžiami į eilės priekį.

Užduotis

- Sukurkite banko operacijų valdymo simuliaciją, kur kiekvienas klientas gali atlikti vieną iš šių veiksmų:
  - Atlikti transakciją (pridėti į eilę/steaką/deklą).
  - Atšaukti transakciją (pašalinti iš eilės/steako/deklo).
  - Peržiūrėti pirmąjį/paskutinį klientą (priklausomai nuo struktūros).
- Simuliacija Python ir C++:
  - Naudokite eilę, steką ir deklą, kad pamatytumėte, kaip kiekviena struktūra keičia aptarnavimo tvarką ir prioritetus.
  - Įgyvendinkite tas pačias funkcijas visose trijose struktūrose, kad stebėtumėte skirtumus.

Tikslas:

- Palyginti, kaip kiekviena duomenų struktūra (eilė, stekas, deklas) keičia klientų aptarnavimo tvarką ir kaip jų naudojimas gali būti pritaikytas realiose situacijose.

Refleksija

- Pagalvokite kaip skirtingos duomenų struktūros keičia aptarnavimo tvarką ir elgesį su duomenimis (FIFO, LIFO, ir dvipusės operacijos su deklu)

Pabaiga