

情報処理 III

後期 第6回課題

4D38 宮里 孝希

2024/11/28

プログラム

linked_list.py

```
class Node:
    def __init__(self, data, next_node=None):
        self.data = data
        self.next = next_node

class LinkedList:
    def __init__(self, head=None):
        self.head = head

    def append(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
        else:
            next_node = self.head
            while next_node.next is not None:
                next_node = next_node.next
            next_node.next = new_node

    def pop(self):
        if self.head is None:
            return None
        current = self.head
        if current.next is None:
            self.head = None
            return current.data
        while current.next.next is not None:
            previous = current
            current = current.next
        previous.next = None
        return current.data

    def pop(self):
        if self.head is None:
            return None
        current = self.head
        if current.next is None:
            self.head = None
            return current.data
        while current.next.next is not None:
            previous = current
            current = current.next
        previous.next = None
        return current.data

    def insert(self, target_data, new_data):
        current = self.head
        while current is not None:
```

```

        if current.data == target_data:
            new_node = Node(new_data, current.next)
            current.next = new_node
            return
        current = current.next
    print(f"Error: {target_data}が見つかりません")

# ノードの削除
def delete(self, target_data):
    current = self.head
    previous = None
    while current is not None:
        if current.data == target_data:
            if previous is None: # 最初のノードを削除
                self.head = current.next
            else:
                previous.next = current.next
            return
        previous = current
        current = current.next
    print(f"Error: {target_data}が見つかりません")

def concat(self, other_list):
    if self.head is None:
        self.head = other_list.head
        return
    current = self.head
    while current.next is not None:
        current = current.next
    current.next = other_list.head

def search(self, target_data):
    current = self.head
    index = 0
    while current is not None:
        if current.data == target_data:
            return index
        current = current.next
        index += 1
    return -1

```

main.py

```
from linked_list import LinkedList
# リストの作成

list1 = LinkedList()
list1.append(1)
list1.append(2)
list1.append(3)

# ノードの挿入
list1.insert(2, 2.5)

# ノードの削除
list1.delete(3)

# ノードの検索
index = list1.search(2.5)
print(f"2.5を含むノードのindex: {index}")

# リストの結合
list2 = LinkedList()
list2.append(4)
list2.append(5)
list1.concat(list2)

# リストの内容を表示
current = list1.head
while current:
    print(current.data)
    current = current.next
```

実行結果

2.5を含むノードのindex: 2

1
2
2.5
4
5
