

Projekt 1 PAMSI

Łukasz Chwistek 243662

14 maj 2021

1. Wstęp

Grupa lab: E12-99I

Termin zajęć: PN 18:55 - 20:35

Prowadzący: dr inż. Piotr Ciskowski

2. Zadanie

Celem zadania jest napisanie programu, który implementuje stos. Do wykonania zadania wykorzystano szablony oraz zaimplementowano listę. Stos zaimplementowano na podstawie tablicy oraz listy jednokierunkowej.

3. Klasa stosu opartego na tablicy

```
/*Definition of StackTab object*/
/*template defines what type and size the object is*/
template <typename T>
class StackTab
{
private:
    T* S; ..... //stack
    unsigned int size; ..... //table size
    unsigned int top_ptr; ..... //top pointer

public:
    StackTab(unsigned int size); ..... //construcor
    ~StackTab(); ..... //destructor
    StackTab(const StackTab& otherStackTab);

    bool isEmpty() const; ..... //boolean method checking if tab is empty
    void print(); ..... //method writing out all current elements on stack
    unsigned int curSize(); ..... //method returning current number of elements on stack
    unsigned int maxSize(); ..... //method returning current maximal size of stack tab
    void expand(); ..... //method expanding
    void push(T data); ..... //adds element on list
    T pop(); ..... //returns element from tab, deletes from tab
    T top(); ..... //returns latest element on tab
    void clear(); ..... //pops whole stack, leaves it empty

    friend void operator++ (StackTab& myStackTab);
    friend void operator-- (StackTab& myStackTab);

    friend std::ostream& operator<< (std::ostream& out, const StackTab& myStackTab);
};
```

4. Klasa stosu opartego na liście

```
8   template <typename T>
9   class StackList
10  {
11  private:
12      struct Node { //element on node struct
13      {
14          T data;
15          Node* next;
16      };
17
18      unsigned int num; //contains number of elements on stack
19      Node* LS; //node to elements on list, using it as stacks top element
20
21  public:
22      StackList(); //constructor
23      ~StackList(); //destruktor
24      bool isEmpty(); //checks if list is isEmpty
25      int size(); //number of elements on list
26      void push(T x); //adds element to list
27      T pop(); //removes element from list
28      void print(); //method writing out all elements on current stack
29      T top(); //returns top of stack, lastest element, without removing it
30      void clear(); //pops the whole stack
31  };
32
```

Reszta kodu dostępna jest na Githubie (https://github.com/Lukkai/PAMSI_1_Stack/tree/master/Stosy) lub dołączonym skompresowanym archiwum rar.

5. Wyniki

Program przeszedł testy, gdzie wywnioskowano, że spełnia swoje zadanie i działa prawidłowo.

6. Wnioski

Przedstawione zadanie można rozwinąć o kolejne działania oraz w celu poprawienia złożoności czasowej struktury ADT.