Министерство образования Республики Беларусь
Учреждение образования
“Брестский Государственный технический университет”
Кафедра ИИТ

Лабораторная работа №7
По дисциплине “Проектирование программного обеспечения интеллектуальных систем”
Тема: “Шаблоны”

Выполнил:
Студент 2 курса
Группы ИИ-21
Кирилович А. А.
Проверил:
Монтик Н. С.

Брест 2022

Цель работы:

1. Изучение правил написания шаблонов функций.
2. Изучение правил написания шаблонов классов.
3. Реализация шаблонов функций.
4. Реализация шаблонов классов.

Ход работы: Разработать три пользовательских класса в соответствии с целью.

```
#include <vector>
#include <string>
#include <iostream>

template <typename T>
class stack {
private:
    std::vector <T> array;
    int head = -1;
public:
    stack() {
        array.resize(0);
    }
    stack(int size) {
        array.resize(size);
    }
    stack(const stack& other) {
        array = other.array;
        head = other.head;
    }
    ~stack() {}
    void push(T x) {
        head++;
        array.push_back(x);
    }
    void pop() {
        head--;
        array.pop_back();
    }
    T front() {
        return array[head];
    }
    bool is_empty() {
        return head == -1;
    }
    void print() {
        int l = array.size();
        for (int i = 0; i < l; i++) {
            std::cout << array[i];
        }
        std::cout << std::endl;
    }
    T operator[](int i) {
        return array[array.size() - i - 1];
    }
};

template <typename T>
class queue {
private:
    std::vector <T> array;
    int tail = 0;
public:
    queue() {
        array.resize(0);
    }
    queue(int size) {
        array.resize(size);
    }
    queue(const queue& other) {
        array = other.array;
        tail = other.tail;
    }
    ~queue() {}
    void push(T x) {
        array.push_back(x);
        tail++;
```

```
    }
    void pop() {
        array.erase(array.cbegin());
        tail--;
    }
    T front() {
        return array[0];
    }
    bool is_empty() {
        return tail == 0;
    }
    void print() {
        int l = array.size();
        for (int i = 0; i < l; i++) {
            std::cout << array[i];
        }
        std::cout << std::endl;
    }
    T operator[](int i) {
        return array[i];
    }
};

template <typename T>
class list {
private:
    struct node {
        T value;
        node* next;
    };
    node* head = nullptr;
    int size = 0;

public:
    list() {
        head = nullptr;
        size = 0;
    }
    list(int size) {
        head = nullptr;
        this->size = size;
    }
    list(const list& other) {
        head = other.head;
        size = other.size;
    }
    ~list() {
        node* current = head;
        while (current != nullptr) {
            node* next = current->next;
            delete current;
            current = next;
        }
    }
    void push(T x) {
        node* new_node = new node;
        new_node->value = x;
        new_node->next = head;
        head = new_node;
        size++;
    }
    void pop() {
        node* temp = head;
        head = head->next;
        delete temp;
        size--;
    }
    T front() {
        return head->value;
    }
    bool is_empty() {
        return size == 0;
    }
    void print() {
        node* temp = head;
        while (temp != nullptr) {
            std::cout << temp->value;
```

```
            temp = temp->next;
        }
        std::cout << std::endl;
    }
    T operator[](int i) {
        node* temp = head;
        for (int j = 0; j < i; j++) {
            temp = temp->next;
        }
        return temp->value;
    }
};

int main() {
    stack <int> s;
    s.push(1);
    s.push(2);
    s.push(3);
    s.print();
    s.pop();
    s.print();
    std::cout << s.front() << std::endl;
    std::cout << s[1] << std::endl;
    std::cout << s.is_empty() << std::endl;

    queue <std::string> q;
    q.push("Hello");
    q.push("World");
    q.push("!");
    q.print();
    q.pop();
    q.print();
    std::cout << q.front() << std::endl;
    std::cout << q[1] << std::endl;
    std::cout << q.is_empty() << std::endl;

    list <char> l;
    l.push('a');
    l.push('b');
    l.push('c');
    l.print();
    l.pop();
    l.print();
    std::cout << l.front() << std::endl;
    std::cout << l[1] << std::endl;
    std::cout << l.is_empty() << std::endl;
}
```

```
123
12
2
1
0
HelloWorld!
World!
World
!
0
cba
ba
b
a
0
```