Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №1**по дисциплине «Структуры и организация данных»  
на тему «Линейные структуры данных»  
часть 2 «Стеки, очереди, деки»  
  
вариант 25

Выполнил:  
Студент Фокин В. К.  
Группа И924Б  
  
Преподаватель:  
Гладевич А. А.

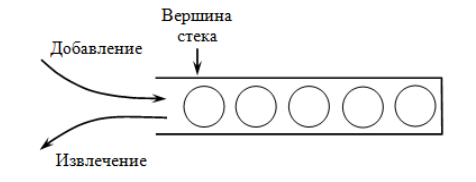
Санкт-Петербург  
2023 г.

Структура данных – стек.

Уровень сложности – ***Средний.***

Написать на языке С++ программу, решающую поставленную в вариативной части задачу, используя указанную там же структуру данных. Выполнение задания осуществляется в два этапа. На первом этапе требуется реализовать необходимую структуру данных с помощью одной структуры хранения (векторной или связной), реализацию оформить в виде класса. Методы класса должны выполнять только основные и 2 дополнительные операции, допустимые над элементами описываемой СД (перечислены ниже). Никаких других операций в классе быть не должно. На втором этапе требуется использовать объекты созданного класса при решении поставленной задачи.

Схематичное изображение структуры данных:

**

Схематичное изображение структуры хранения, использованной в программе:

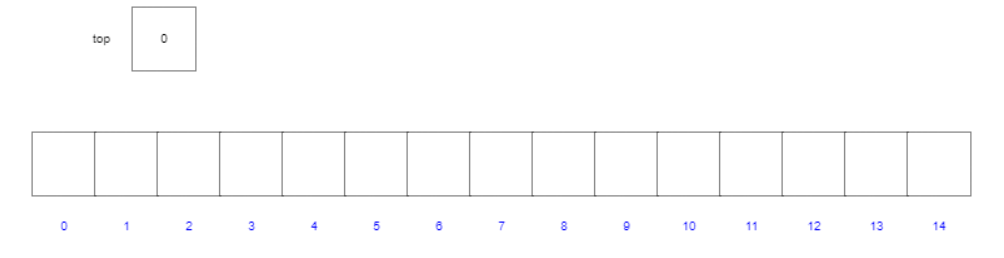
**

Рисунок 1 – Пустой стек

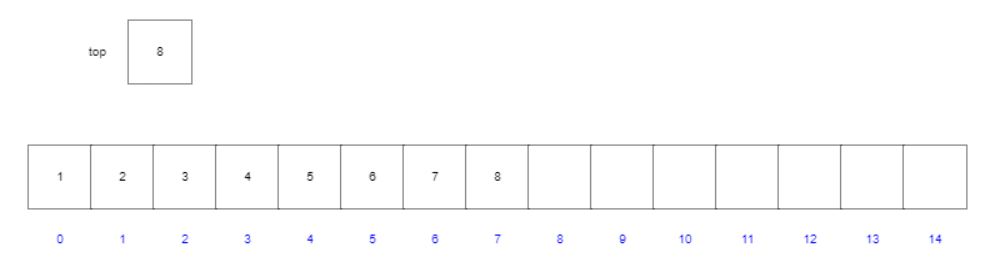


Рисунок 2 – Частично заполненный стек

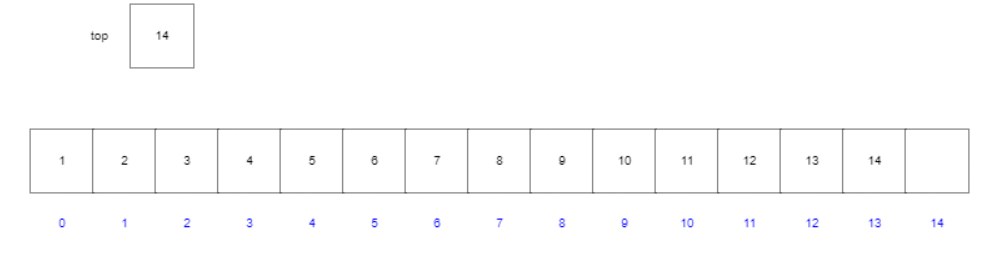


Рисунок 3 – Заполненный стек

Текст программы:

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <iomanip>

#include <windows.h>

using namespace std;

const int MAX = 50;

//структура блинчик

struct pancake {

string picture;

int number;

};

//стек

class Stack {

pancake data[MAX];

int ind;

public:

Stack() {

ind = -1;

}

// конструктор копирования

Stack(const Stack& other) {

ind = other.ind;

for (int i = 0; i <= ind; i++) {

data[i] = other.data[i];

}

}

// перегруженный оператор присваивания

Stack& operator=(const Stack& other) {

if (this == &other) {

return \*this;

}

ind = other.ind;

for (int i = 0; i <= ind; i++) {

data[i] = other.data[i];

}

return \*this;

}

void push(pancake value) {

if (ind == MAX - 1) {

cout << "Stack overflow" << endl;

return;

}

ind++;

data[ind] = value;

}

pancake pop() {

if (ind == -1) {

cout << "Stack underflow" << endl;

return pancake{ "", -1 };

}

pancake top = data[ind];

ind--;

return top;

}

pancake top() {

if (ind == -1) {

cout << "Stack is empty" << endl;

return pancake{ "", -1 };

}

return data[ind];

}

bool empty() {

return ind == -1;

}

bool full() {

return ind == MAX - 1;

}

};

int random(int min, int max) {

return min + rand() % (max - min + 1);

}

int main() {

SetConsoleOutputCP(1251);

srand(time(nullptr));

int numofchildren, total = 0, counter = 24;

do {

cout << "----------------------------------" << endl;

cout << "Введите количество внуков (2 - 4)" << endl;

cin >> numofchildren;

if (numofchildren < 2) {

cout << "Мало детей.." << endl;

}

else if (numofchildren > 4) {

cout << "Слишком много детей" << endl;

}

} while (numofchildren < 2 || numofchildren > 4);

cout << "----------------------------------" << endl;

system("cls");

Stack s1;

cout << "-------------------------------------------" << endl;

cout << "| Номер блина | Картинка | Еще испечь |" << endl;

cout << "-------------------------------------------" << endl;

for (int i = 0; i < 25; i++) {

string pictures[] = { "цветочки", "грибки", "котятки" };

int picInd = random(0, 2);

string picture = pictures[picInd];

pancake panc = { picture, i };

s1.push(panc);

cout << "| " << left << setw(8) << i + 1 << "| " << left << setw(10) << picture << "| " << left << setw(7) << 49 - i << "|" << endl;

}

cout << "-------------------------------------------" << endl;

int p = 25;

for (int i = 0; i < numofchildren; i++) {

int toEat = random(8, 15);

int eaten = 0;

cout << "\nВнук" << i + 1 << " может съесть " << toEat << " блинчиков" << endl;

int t = toEat;

cout << "--------------------------------------------------------------" << endl;

cout << "| Прошло сек | Cъедено | Испечено | Еще испечь |" << endl;

cout << "--------------------------------------------------------------" << endl;

for (int j = 1; j <= t; j++) {

if (j % 2 == 1) {

pancake panc = s1.top();

if (s1.empty()) {

cout << "Блинчики кончились (((" << endl;

break;

}

else {

if (panc.number - 9 > 0) {

cout << "| " << left << setw(7) << j \* 30 << "| #" << panc.number + 1 << ", " << left << setw(10) << panc.picture << "| -- | ";

}

else {

cout << "| " << left << setw(7) << j \* 30 << "| #" << panc.number + 1 << ", " << left << setw(11) << panc.picture << "| -- | ";

}

if (p - 9 > 0) {

cout << left << setw(7) << p << "|" << endl;

}

else {

cout << left << setw(7) << p << "|" << endl;

}

s1.pop();

eaten += 1;

total += 1;

}

}

else {

pancake panc = s1.top();

if (s1.empty()) {

cout << "Блинчики кончились (((" << endl;

break;

}

else {

if (panc.number - 9 > -1) {

cout << "| " << left << setw(7) << j \* 30 << "| #" << panc.number + 1 << ", " << left << setw(10) << panc.picture;

}

else {

cout << "| " << left << setw(7) << j \* 30 << "| #" << panc.number + 1 << ", " << left << setw(11) << panc.picture;

}

s1.pop();

eaten += 1;

total += 1;

string pictures[] = { "цветочки", "грибки", "котятки" };

int picInd = random(0, 2);

string picture = pictures[picInd];

counter += 1;

panc = { picture, counter };

s1.push(panc);

if (p - 10 > -1) {

cout << "| #" << counter + 1 << ", " << left << setw(9) << picture << "| " << left << setw(7) << p - 1 << "|" << endl;

}

else {

cout << "| #" << counter + 1 << ", " << left << setw(9) << picture << "| " << left << setw(7) << p - 1 << "|" << endl;

}

p -= 1;

}

}

}

if (eaten == 0) {

cout << "--------------------------------------------------------------" << endl;

cout << "Внук" << i + 1 << " не съел ни одного блинчика :(\n" << endl;

}

else if (eaten < toEat) {

cout << "--------------------------------------------------------------" << endl;

cout << "Внуку" << i + 1 << " не хватило " << toEat - eaten << " блиничиков, чтобы наесться ;(\n" << endl;

}

else {

cout << "--------------------------------------------------------------" << endl;

cout << "Внук" << i + 1 << " съел " << eaten << " блинчиков и больше не хочет ;)\n" << endl;

}

system("pause");

}

cout << "\n----------------------------------" << endl;

cout << "Всего было съедено " << total << " блинчиков" << endl;

if (50 - total > 0) {

cout << "Еще остлось " << 50 - total << " блинчиков" << endl;

cout << "----------------------------------" << endl;

}

else {

cout << "Блинчиков больше не осталось.." << endl;

cout << "----------------------------------" << endl;

}

}

Результаты работы программы:

При запуске программа просит ввести количество внуков у бабушки, что показано на рисунке 4.

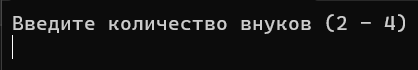


Рисунок 4 – Запуск программы

При вводе значений меньше 2 или больше 4, выведется сообщение об ошибке, что показано на рисунке 5.

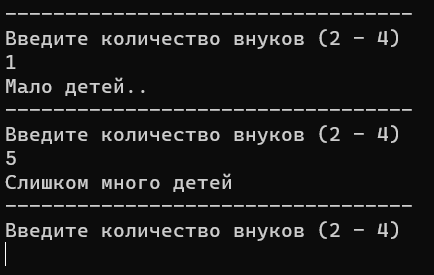


Рисунок 5 – Ввод некорректных значений

После ввода подходящих значений стек частично заполняют 25 объектов. Выводится таблица, в которой указаны номер испеченного блинчика, картинка на этом блинчике и на сколько блинчиков осталось теста, что показано на рисунке 6.

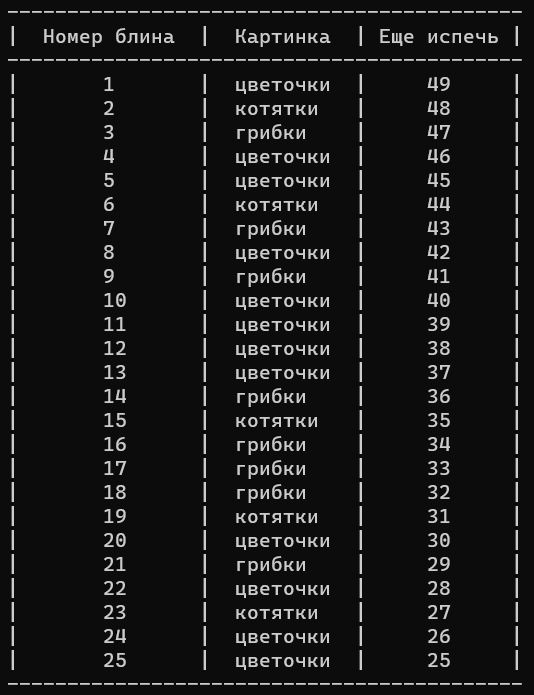


Рисунок 6 – Частичное заполнение стека

Затем выводится сообщение о том, сколько блинчиков может съесть внук. Это значение от 8 до 15 и задается случайно программой. Так как внук может съесть два блинчика за минуту, а бабушка успевает приготовить всего один, из стека сначала удаляется два элемента, а потом добавляется один. Если внук съест столько блинов, сколько задала программа, выведется сообщение об этом, что показано на рисунке 7.

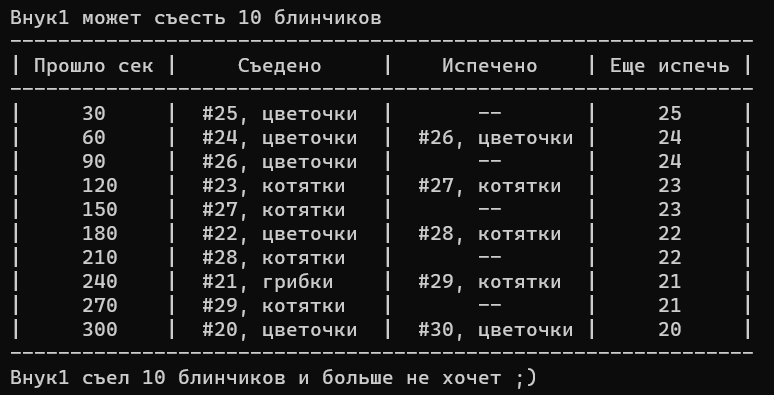


Рисунок 7 – Вывод сообщений и результат работы программы

Если блинчики кончились до того, как внук съел заданное количество, выведется сообщение о том, что стек пустой и сколько блинчиков не хватило внуку, что показано на рисунке 8.

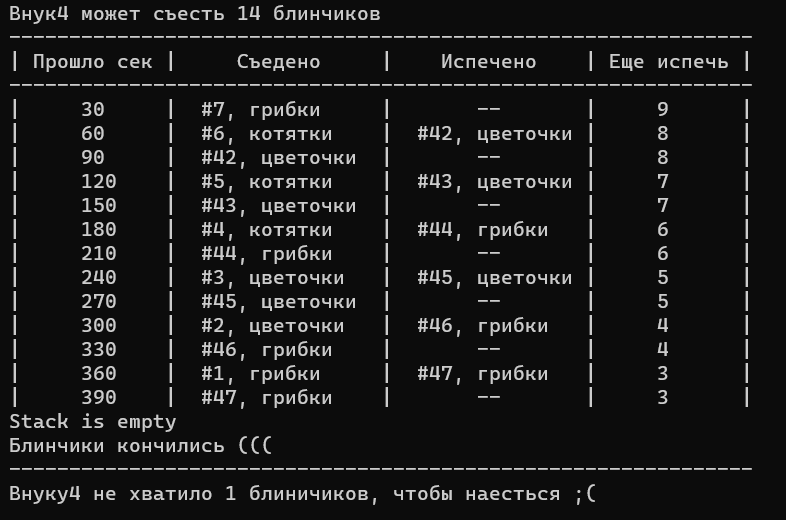


Рисунок 8 – Вывод сообщения о пустом стеке

Процесс повторяется для оставшихся внуков.

В конце программы выводится сообщение о том, сколько блинчиков было съедено и сколько блинчиков осталось, что показано на рисунке 9.

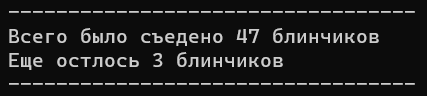


Рисунок 9 – Завершение программы