**Цель работы:** изучение принципов организации и программирование основных операций, выполняемых с линейными списковыми структурами.

**Постановка задачи:** разработать программу формирования стека. В программе предусмотреть вывод элементов и возврат количества элементов.

**Описание переменных и методов**

В программе использовалось следующие переменные:

**private** **byte**[] stack – переменная с непосредственно стеком

**private** **int** size – переменная, хранящая количество элементов стека

**public** **int** returnStackSizeValue() – метод, возвращающий количество элементов стека

**public** **byte**[] showLeftData(**int** size) – метод, выводящий оставшиеся элементы стека

**public** **boolean** push(**byte** data) – метод, записи данных в стек

**public** **byte** pop(**boolean**[] empty) – метод, извлечения данных из стека

**Текст программы.**

**package** com.worm2fed.kursach\_prog;

**public** **class** Stack {

// переменная для будущего стека

**private** **byte**[] stack;

// переменная для размера стека

**private** **int** size;

// стек по умолчанию

**public** Stack() {

// создаём стек размером в 16 ячеек.

**this**(16);

}

**public** Stack(**int** n) {

// создаём стек размером в n ячеек.

stack = **new** **byte**[n];

size = 0;

}

// функция для записи данных в стек

**public** **boolean** push(**byte** data) {

**if** (size == st.length)

**return** **false**;

**else**

**if** (size != 0)

**for** (**int** i = (size - 1); i >= 0; i--)

stack[i + 1] = stack[i];

stack[0] = data;

size++;

**return** **true**;

}

}

// функция для извлечения данных из стека

**public** **byte** pop(**boolean**[] empty) {

**if** (size == 0) {

empty[0] = **true**;

**return** (**byte**) -128;

}

**else** {

**byte** data = stack[0];

**for** (**int** i = 0; i < (size - 1); i++)

stack[i] = stack[i + 1];

size--;

**return** data;

}

}

**public** **int** returnStackSizeValue() {

**return** size;

}

**public** **byte**[] showLeftData(**int** size) {

// **TODO** Auto-generated method stub

**byte**[] left\_mem = **new** **byte**[size];

**for** (**int** i = 0; i < size; i++)

left\_mem[i] = stack[i];

**return** left\_mem;

}

}

**Вывод:** Изучили принципы работы с базовыми линейными структурами