Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления
Кафедра — интеллектуальных информационных технологий

К защите допустить:

Заведующий кафедрой ИИТ
Д.В. Шункевич

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к лабораторной работе
на тему:

Динамическая структура СТЕК

Студент гр. 321701 А. С. Астахов
Руководитель С.И. Матюшкин

Минск 2024

Цель: изучить алгоритмы работы с динамическими структурами в виде стека.
Задача: составить и отладить программу.

Индивидуальное задание

Написать программу по созданию, добавлению, просмотру и решению приведенных далее задач (в рассмотренных примерах это действие отсутствует) для однонаправленного линейного списка типа *Stack*. Реализовать сортировку стека методами, рассмотренными в подразделе 3.1.

Решение поставленной задачи описать в виде блок-схемы.

Во всех заданиях создать список из положительных и отрицательных случайных целых чисел.

1. Разделить созданный список на два: в первом – положительные числа, во втором – отрицательные.

```
struct Stack {
    int info;
    Stack* next;
} *start, *positive, *negative;


Stack *inStack(Stack *p, int in) {
    Stack *t = new Stack;
    t -> info = in;
    t -> next = p;

    return t;
}

void viewWholeStack(Stack* p) {
    Stack *t = p;

    while (t != nullptr) {
        cout << t -> info << endl;
        t = t -> next;
    }
}

Stack* outStack(Stack *p, int *out) {
    Stack* t = p;
    *out = p -> info;

    p = p -> next;

    delete t;
    return p;
```

```
}

void deleteAll(Stack* p) {
    Stack *t;
    while (p != nullptr)
    {
        t = p;
        p = p -> next;
        delete t;
    }
}

Stack *createPositive(Stack* p) {
    Stack *stack = new Stack;
    deleteAll(stack);
    Stack *t = p;

    while (t != NULL) {
        if (t -> info > 0)
            stack = inStack(stack, t -> info);
        t = t -> next;
    }

    return stack;
}

Stack *createNegative(Stack* p) {
    Stack *stack = new Stack;
    deleteAll(stack);
    Stack *t = p;

    while (t != NULL) {
        if (t -> info < 0)
            stack = inStack(stack, t -> info);
        t = t -> next;
    }

    return stack;
}

void Sort_p(Stack **p) {
    Stack *t = NULL, *t1, *r;
    if ((*p) -> next -> next == NULL)
    return;
    do {
        for (t1=*p; t1-> next->next != t; t1=t1-> next)
            if (t1->next->info > t1-> next-> next-> info){
                r = t1->next->next;
                t1 -> next -> next = r -> next;
                r-> next =t1-> next;
                t1-> next = r;
            }
        t = t1-> next;
    } while ((*p)-> next -> next != t);
}
```

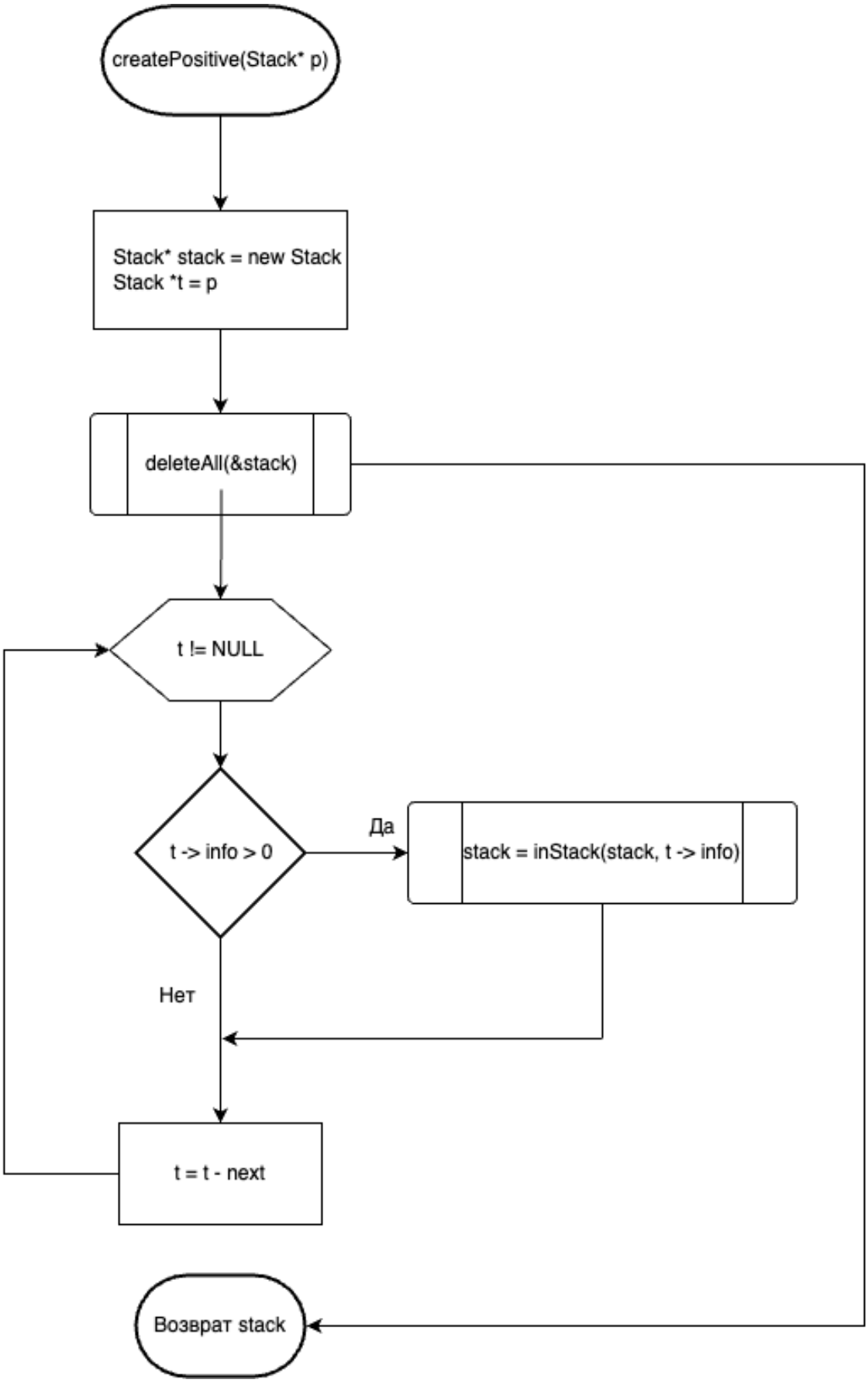
```
int main(int argc, const char * argv[]) {
    int choise, in, localeChoise;

    for( int i = 0; i < 9; i++)
    {
        start = inStack(start, rand() % 10 - 5);
    }

    while (true) {
        cout << "Веберете операцию над стеком" << endl;
        cout << "1 - Добавление элемента" << endl;
        cout << "2 - Просмотр стека" << endl;
        cout << "3 - Нахождение всех отрицательных значений" << endl;
        cout << "4 - Наождение всех поолжительных значений" << endl;
        cout << "5 - Очистка стека" << endl;
        cout << "6 - Сортировка стека" << endl;
        cout << "7 - Выход" << endl;

        cin >> choise;

        switch (choise) {
            case 1:
                cout << "Ввежите желаемое значение для добавления в
стек" << endl;
                cin >> in;

                start = inStack(start, in);
                break;

            case 2:
                cout << "Вывести первоначальную(1), положительную(2) или
отрицательную(3) очередь?" << endl;
                cin >> localeChoise;

                switch (localeChoise) {
                    case 1:
                        viewWholeStack(start);
                        break;

                    case 2:
                        viewWholeStack(positive);
                        break;

                    case 3:
                        viewWholeStack(negative);
                        break;

                    default:
                        break;
                }
                break;

            case 3:
                negative = createNegative(start);
                break;
```

```
            case 4:
                positive = createPositive(start);
                break;

            case 5:
                cout << "Какой стек очистить? (1 – первоначальный, 2 –
стек положительных значений, 3 – стек отрицательных значений)" << endl;
                int tempChoise;
                cin >> tempChoise;
                switch (tempChoise) {
                    case 1:
                        deleteAll(start);
                        break;

                    case 2:
                        deleteAll(positive);
                        break;

                    case 3:
                        deleteAll(negative);
                        break;

                    default:
                        break;
                }
                break;
            case 6:
                Sort_p(&start);
                break;
            case 7:
                return 0;

            default:
                break;
        }
    }
}
```

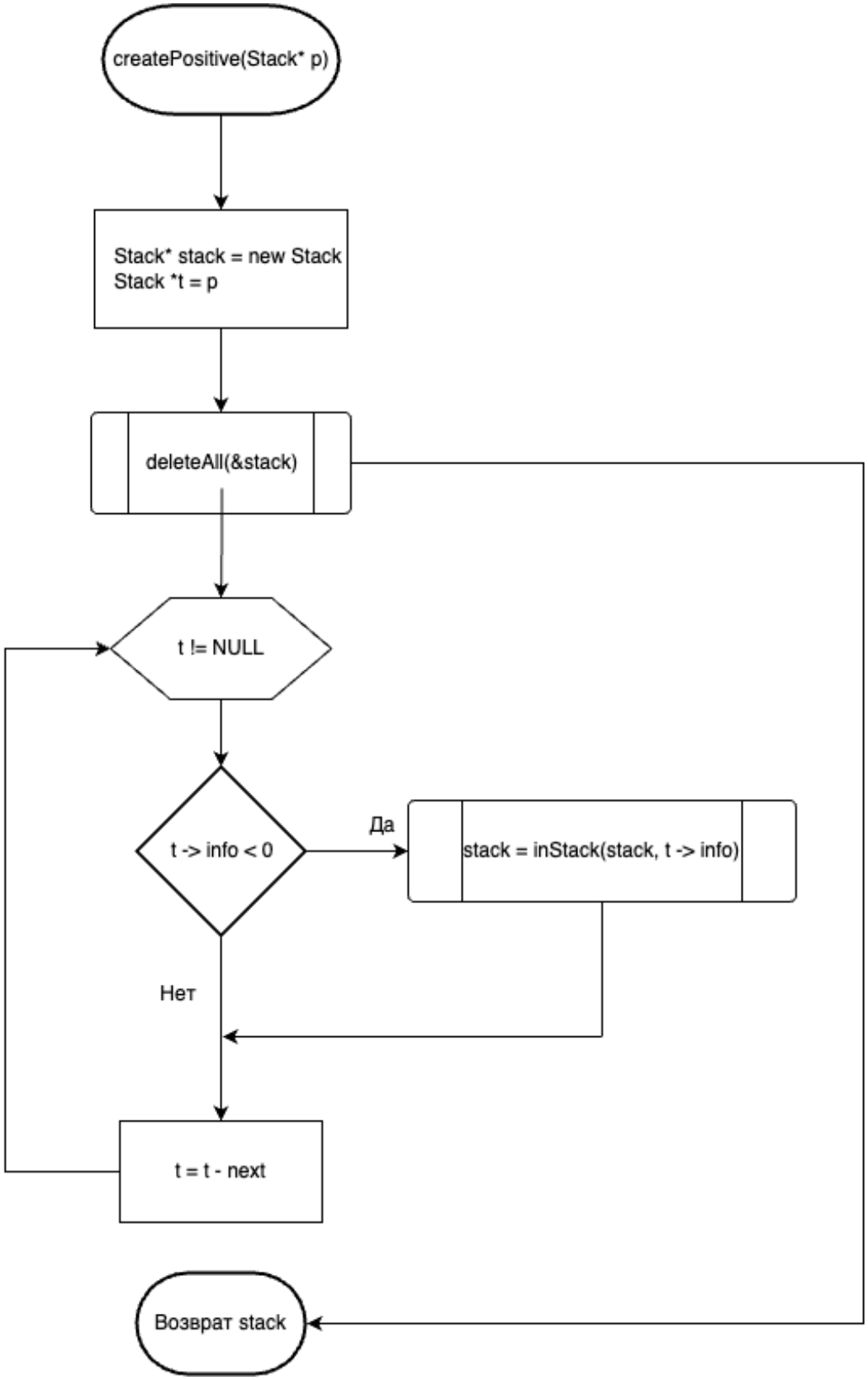
Результат выполнения:

```
Вывести первоначальную(1), положительную(2) или отрицательную(3) очередь?1

-2
3
-1
-3
-5
3
-2
4
2
Веберете операцию над стеком
1 - Добавление элемента
2 - Просмотр стека
3 - Нахождение всех отрицательных значений
4 - Наождение всех поолжительных значений
5 - Очистка стека
6 - Выход
2
Вывести первоначальную(1), положительную(2) или отрицательную(3) очередь?
3
-2
-5
-3
-1
-2
Веберете операцию над стеком
1 - Добавление элемента
2 - Просмотр стека
3 - Нахождение всех отрицательных значений
4 - Наождение всех поолжительных значений
5 - Очистка стека
6 - Выход
2
Вывести первоначальную(1), положительную(2) или отрицательную(3) очередь?
2
2
4
3
3
```

Блок-Схема функции нахождения положительных значений:



Блок-Схема функции нахождения отрицательных значений:



Вывод: изучил алгоритмы работы с динамическими структурами в виде стека, составил и отладил программу.