

Цель работы: познакомиться со способами описания алгоритма с использованием блок-схем и псевдокода.

Вариант 4

Задание:

Алгоритм заказа товаров из интернет-магазина.

1. Псевдокод

Создадим синтаксис для моего псевдокода, чтобы понимать какие действия совершаются.

Большими русскими буквами будут, являются операторами, чтобы легче было отделять и читать.

1) Создание алгоритма, вводимые аргументы разделяются через запятую:

```
АЛГ <имя алгоритма>(<аргумент 1>, <аргумент 2>, ..., <аргумент N>) {  
  <цепочка действий>  
}
```

2) Блок с действиями для отделения операций и содержания действий:
{ <цепочка действий> }

3) Прерывание блока с действиями, в котором находится:
ВЫХОД

4) Каждое действие разделяется следующей строкой:

```
<действие 1>  
<действие 2>  
...  
<действие N>
```

5) Оператор, возвращающий значения алгоритма:

ВЕРНУТЬ <значение>

6) Получить значение алгоритма:

<имя алгоритма>(<аргумент 1>, <аргумент 2>, ..., <аргумент N>)

7) Создание условия:

Краткая:

```
ЕСЛИ <условие> {  
  <цепочка действий>  
}
```

Полная:

```
ЕСЛИ <условие> {  
  <цепочка действий>  
}  
ИНАЧЕ ЕСЛИ <условие> {  
  <цепочка действий>  
}  
ИНАЧЕ {  
  <цепочка действий>  
}
```

8) Цикл перебора:

```
ЦИКЛ <значение> ИЗ <список значений> {  
  <цепочка действий>  
}
```

9) Цикл с постусловием (совершающий сначала действие, а после идёт условие):

ДЕЛАЙ {
 <цепочка действий>
}
ПОКА <условие>

10) Инициализация и присваивание:
 <имя переменной> = <значение>

11) Арифметические операторы:
 <значение 1> +, -, *, / <значение 2>
 Сложение(+)
 Вычитание(-)
 Умножение(*)
 Деление(/)

12) Операторы сравнения:
 <значение 1> ==, ≠, <, > <значение 2>
 Равенство(==)
 Неравенство(≠)
 Меньше(<)
 Больше(>)

13) Логические операторы:
 <значение 1> **ИЛИ**, **И** <значение 2>
 НЕ <значение>

14) Создание последовательности значений:
 <имя последовательности> = [<значение 1>, <значение 2>, ..., <значение N>]

15) Добавление значения в последовательность:
 ДОБАВИТЬ <значение> **В** <последовательность>

Псевдокод алгоритма заказа:

Входными данными буду являться интернет-магазин и список покупок, составленный покупателем.

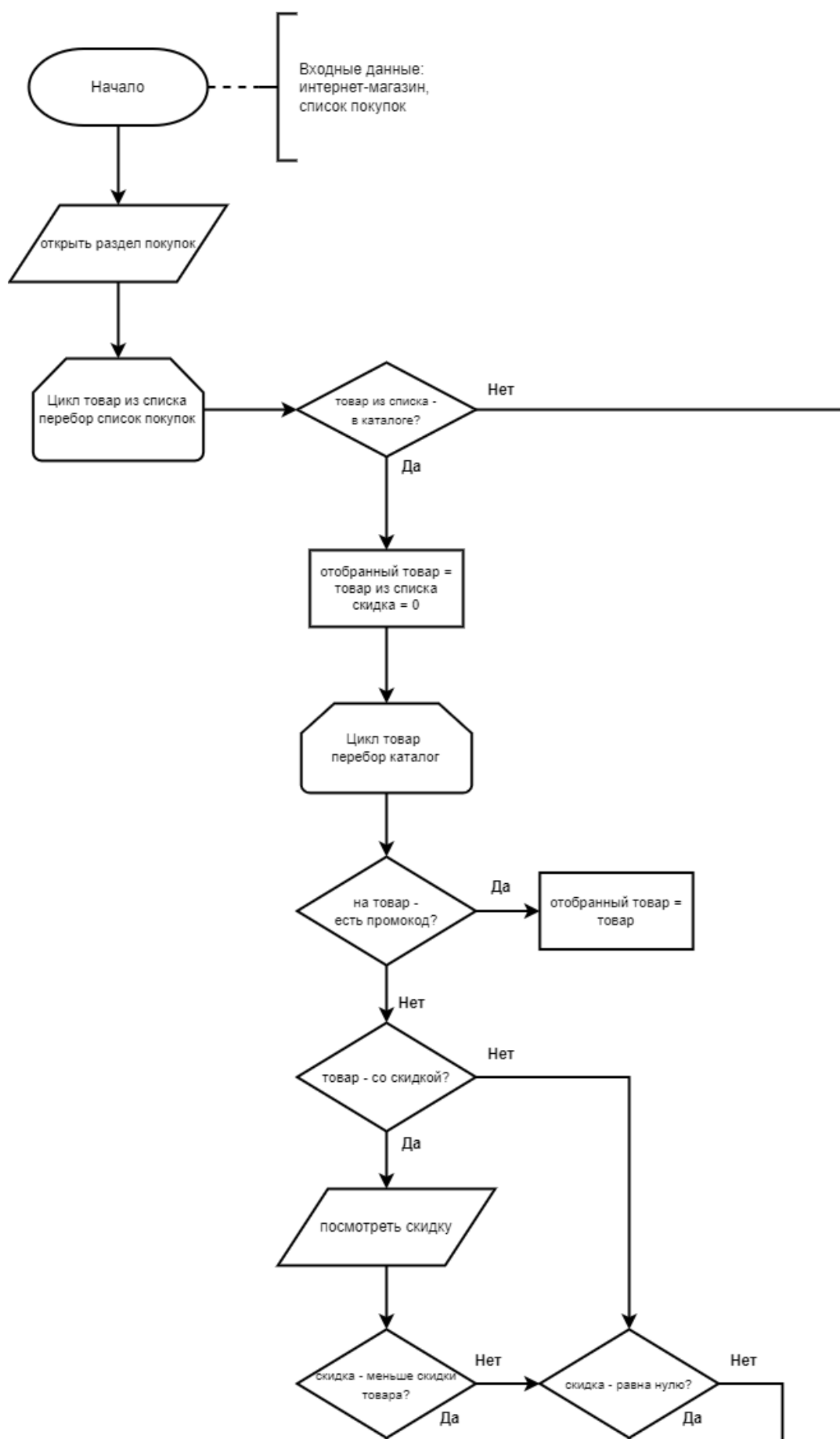
```
АЛГ заказ(интернет-магазин, список_покупок) {  
    корзина = []  
    каталог = открыть_раздел_покупок(интернет-магазин)  
    ЦИКЛ товар_из_списка ИЗ список_покупок {  
        ЕСЛИ товар из списка есть в каталоге {  
            отобранный_товар = товар_из_списка  
            скидка = 0  
            ЦИКЛ товар ИЗ каталог {  
                ЕСЛИ есть промокод на товар {  
                    отобранный_товар = товар  
                    ВЫХОД  
                } ИНАЧЕ ЕСЛИ товар со скидкой {  
                    скидка_товара = посмотреть_скидку(товар)  
                    ЕСЛИ скидка < скидка_товара {  
                        скидка = скидка_товара  
                        отобранный_товар = товар  
                    }  
                } ИНАЧЕ ЕСЛИ скидка == 0 {  
                    отобранный_товар = товар  
                }  
            }  
            ДОБАВИТЬ отобранный_товар В корзина  
        }  
    }  
    тип_оплаты = совершить_заказ(интернет-магазин, корзина,  
    сумма_денег)  
    ДЕЛАЙ {  
        звонок = ждать_звонок()  
    } ПОКА звонок ≠ заказ_прибыл  
    продукты = получить_заказ(тип_оплаты, интернет-магазин, корзина,  
    сумма_денег)
```

```
        ВЕРНУТЬ продукты
    }
```

```
АЛГ совершить_заказ(интернет-магазин, корзина, сумма_денег) {
    ЕСЛИ оплата по прибытии {
        ВЕРНУТЬ оплата не совершена
    } ИНАЧЕ {
        оплатить(интернет-магазин, корзина, сумма_денег)
        ВЕРНУТЬ оплата совершена
    }
}
```

```
АЛГ получить_заказ(тип_оплаты, интернет-магазин, корзина, сумма_денег) {
    ЕСЛИ оплата не совершена {
        оплатить(интернет-магазин, корзина, сумма_денег)
    }
    ВЕРНУТЬ продукты
}
```

2. Блок-схема



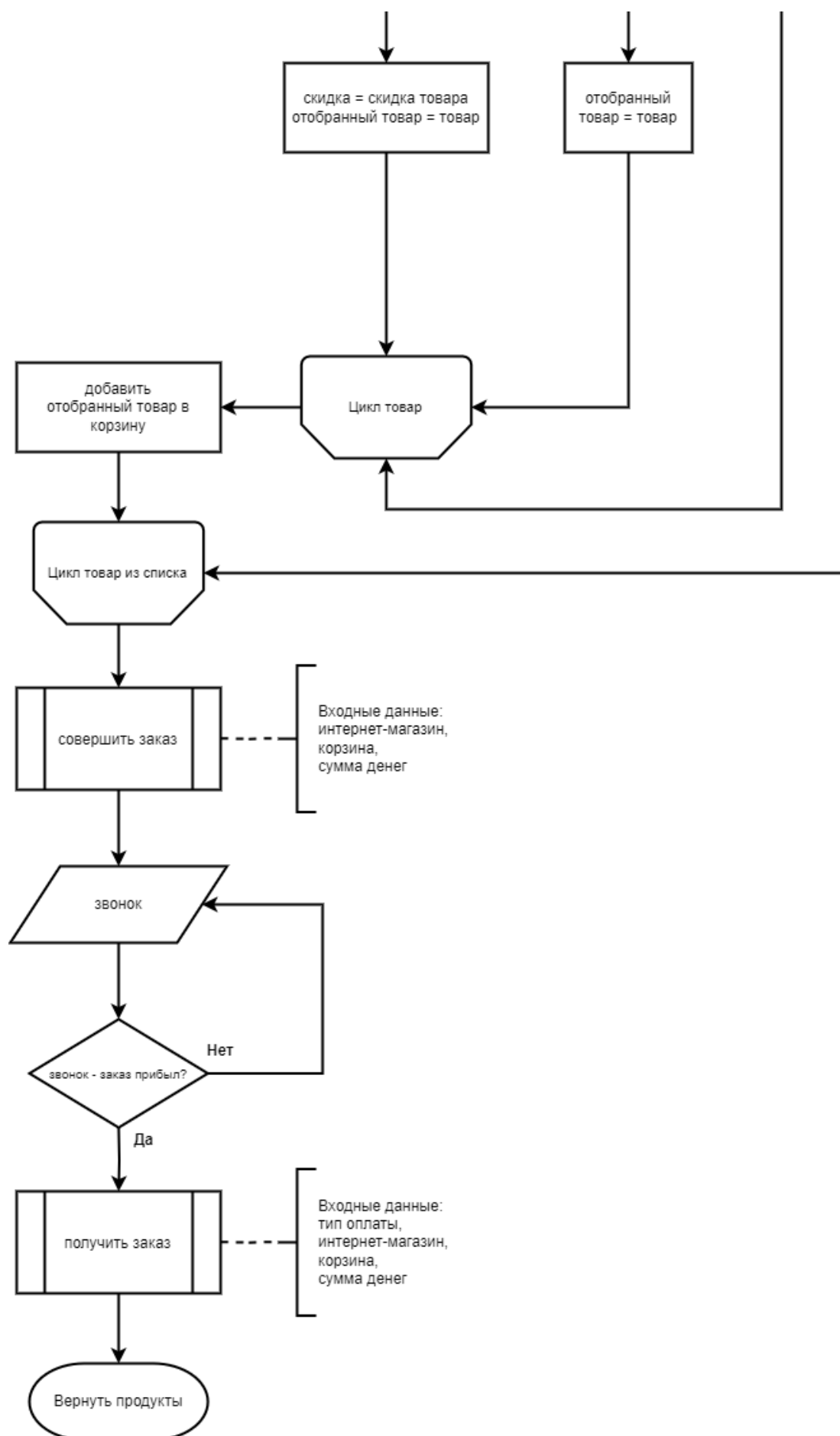


Рис 2.1 – Блок-схема алгоритма “заказ”

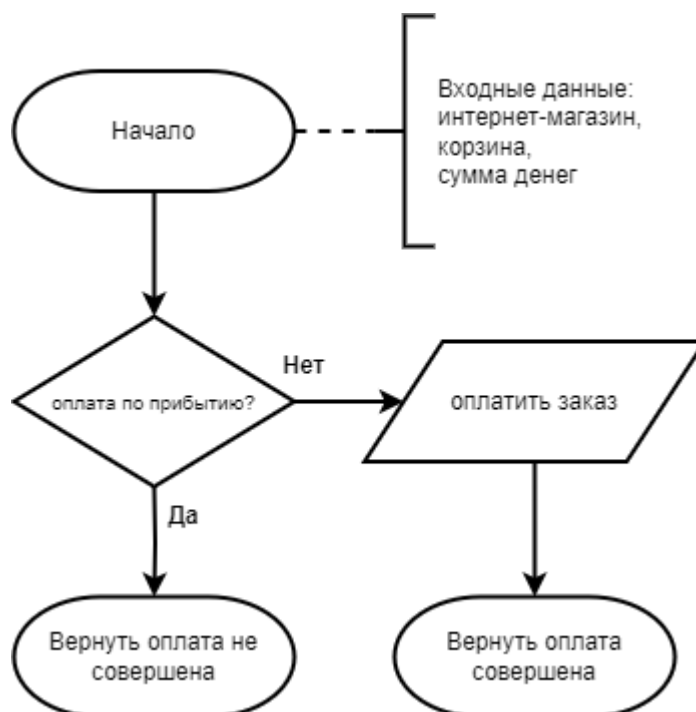


Рис 2.2 – Блок-схема алгоритма “совершить заказ”



Рис 2.3 – Блок-схема алгоритма “получить заказ”

Вывод

Я создала свой простой и понятный псевдокод, а также с помощью него разработала свой алгоритм под задание. Научилась создавать блок-схемы по своему алгоритму.