Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №2**

**«ИЗУЧЕНИЕ ОДНОМЕРНЫХ МАССИВОВ И СТРОК»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАМИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Тетерина Юлия Константиновна

Преподаватель:

Сергеева Елизавета Григорьевна

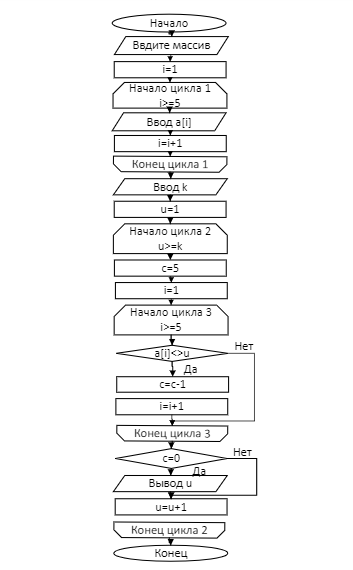
Киров

2022

1. **Цель работы:** получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.
2. **Вариант: 21**

**Задание 1:** написать программу, определяющую какие числа из интервала [1; k] не встречаются в заданном массиве.

1. **Схема алгоритма:**



**Рис. 1 – Схема алгоритма**

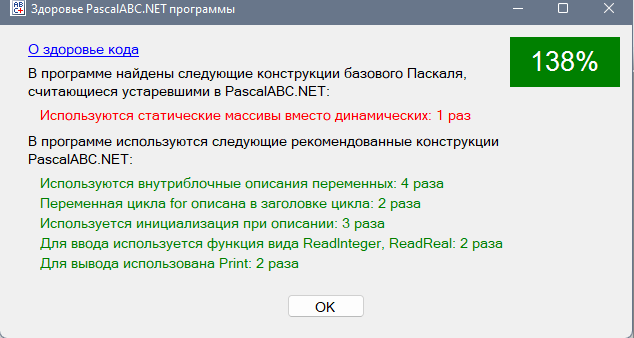
1. **Коды программы:**
2. **begin**
3. **var** a:**array**[1..5] **of** integer;
4. println('введите массив:');
5. **begin**
6. **for var** i:=1 **to** 5 **do**
7. a[i]:=ReadInteger;
8. **end**;
9. **var** k:=ReadInteger('введите k:');
10. **var** u:=1;
12. **while** u<=k **do**
13. **begin**
14. **var** c:=5;
15. **for var** i:=1 **to** 5 **do**
16. **begin**
17. **if** a[i]<>u **then**
18. c:=c-1;
19. **end**;
20. **if** c=0 **then** print(' не используются:',u);
21. u:=u+1;
22. **end**;
23. **end**.
24. **Результат программы:**

Изображение выглядит как текст

Автоматически созданное описание

**Рис. 2 – Результат программы**

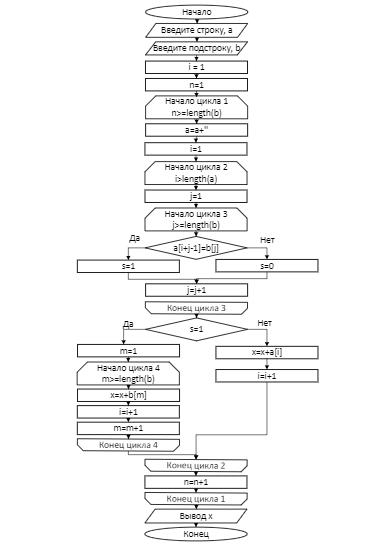
1. **Здоровье кода:**



**Рис. 3 – Здоровье кода**

**Задание 2:** написать программу, выполняющую в строке замену заданной подстроки не инверсию ей.

1. **Схема алгоритма:**



**Рис. 4 – Схема алгоритма**

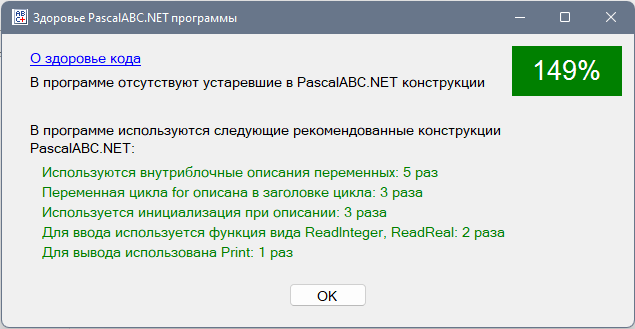
1. **Коды программы:**
2. **begin**
3. **var** a:=ReadString('Введите строку:');
4. **var** b:=ReadString('Введите подстроку:');
5. **var** s:integer;
6. **var** x:string;
7. **var** i:=1;
8. **for var** n:=1 **to** length(b) **do**
9. a:=a+' ';
10. **while** i<length(a) **do begin**
11. **for var** j:=1 **to** length(b) **do begin**
12. **if** a[i+j-1]=b[j] **then** s:=1 **else begin** s:=0; **break**; **end**;
13. **end**;
14. **if** s=1 **then for var** m:=length(b) **downto** 1 **do begin** x:=x+b[m]; i:=i+1;
15. **end**
16. **else begin** x:=x+a[i]; i:=i+1;
17. **end**;
18. **end**;
19. println(x);
20. **end**.
21. **Результат программы:**

Изображение выглядит как текст

Автоматически созданное описание

**Рис. 5 – Результат программы**

1. **Здоровье кода:**



**Рис. 6 – Здоровье кода**

1. **Вывод:** при выполнении данной работы у нас возникли различные вопросы, на которые мы постепенно находили ответы. Первый вопрос, с которым мы столкнулись, «что такое здоровье кода? И как это сделать?». И мы выяснили, что здоровье кода PascalABC.NET - важнейший показатель кода, характеризующий то, используются ли в программе устаревшие конструкции базового Паскаля, или напротив новые конструкции PascalABC.NET, рекомендованные практически для любой программы. Вторым вопросом стало то, как выполнить второе задание, без использования специальных функций? Все оказалось очень просто, необходимо было использовать массив.