**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

**Отчёт**

По дисциплине «Основы программной инженерии»

Выполнил:

Студент 1 курса 10 группы ПИ

Макаревич Кирилл Витальевич

Проверила: Волчек Дарья Ивановна

2024, Минск

**№4**

***Cловесно-формульное описание алгоритма:***

1. **Ввод символа**:

* Пользователь вводит символ с клавиатуры.

2. **Определение категории символа**:

Проверить, является ли введенный символ **цифрой**. Для этого можно использовать условие проверки, находится ли код символа в диапазоне 48-57 (диапазон кодов цифр в таблице ASCII/Windows-1251).

* Проверить, является ли символ **буквой латинского алфавита**. Для этого необходимо проверить два диапазона:
  + Заглавные латинские буквы: диапазон кодов от 65 до 90.
  + Строчные латинские буквы: диапазон кодов от 97 до 122.
* Проверить, является ли символ **буквой русского алфавита**. В кодировке Windows-1251 коды для букв русского алфавита:
  + Заглавные буквы: диапазон от 192 до 223.
  + Строчные буквы: диапазон от 224 до 255.
* Если символ не попадает ни в один из вышеперечисленных диапазонов, считать его **другим символом** (например, знаки препинания, пробелы, специальные символы и т.д.).

3. **Вывод результата**:

* Вывести в консоль, к какой категории относится символ, его значение и его код в соответствующей кодировке (ASCII или Windows-1251).

***Описание алгоритма, используя псевдокод:***

НАЧАЛО

Ввести символ (char)

Получить код символа (code = код символа)

ЕСЛИ code >= 48 И code <= 57 ТО:

Вывести "Символ является цифрой"

ИНАЧЕ ЕСЛИ (code >= 65 И code <= 90) ИЛИ (code >= 97 И code <= 122) ТО:

Вывести "Символ является латинской буквой"

ИНАЧЕ ЕСЛИ (code >= 192 И code <= 223) ИЛИ (code >= 224 И code <= 255) ТО:

Вывести "Символ является русской буквой"

ИНАЧЕ:

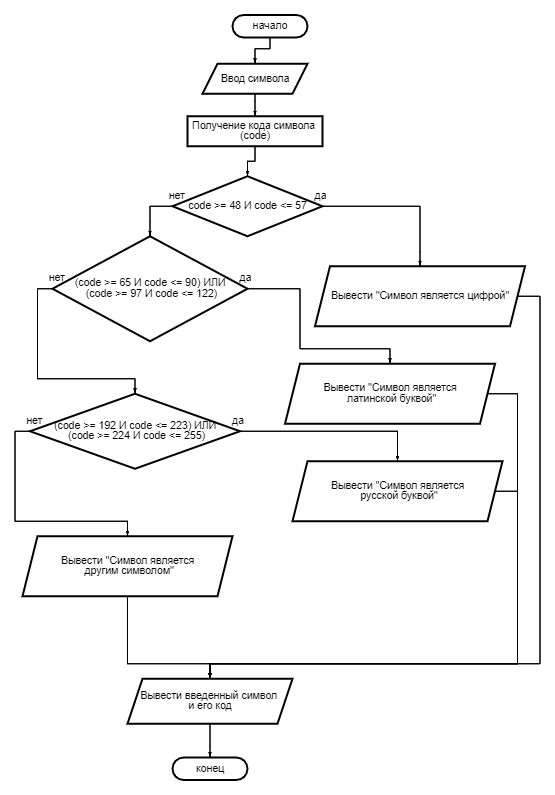
Вывести "Символ является другим символом"

Вывести "Символ: " + char

Вывести "Код символа: " + code

КОНЕЦ

***Блок-схема алгоритма***



**№5**

***Словесно-формульное описание алгоритма:***

1. Вводится фамилия "Иванюк".

2. Из фамилии выбираются первые пять букв: И, В, А, Н, Ю.

3. Для каждой буквы в заглавном и строчном написании находятся их коды в кодировке Windows-1251:

4. Для каждой буквы вычисляется разница между кодом заглавной и строчной версии: разница = код\_прописной - код\_строчной.

5. Результат выводится на экран для каждой из первых пяти букв фамилии.

***Описание алгоритма, используя псевдокод:***

НАЧАЛО

Фамилия = "Иванюк"

ПЕРВЫЕ\_ПЯТЬ\_БУКВ = "Иваню"

ДЛЯ каждой буквы в ПЕРВЫЕ\_ПЯТЬ\_БУКВ:

символ\_прописной = заглавная версия буквы

символ\_строчный = строчная версия буквы

код\_прописной = получить код символа\_прописной

код\_строчный = получить код символа\_строчный

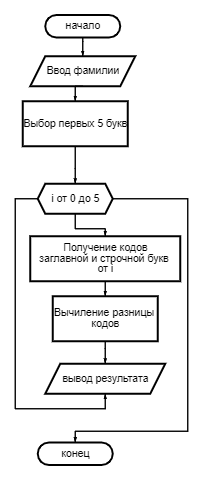
разница = код\_прописной - код\_строчной

Вывести "Буква: " + буква

Вывести "Разница кодов: " + разница

КОНЕЦ

***Блок-схема алгоритма***



**№18**

***Cловесно-формульное описание алгоритма:***

1. Вводится символ, который нужно перевести в строчную версию.

2. Проверяется, принадлежит ли символ диапазону заглавных букв в кодировке Windows-1251. Заглавные русские буквы находятся в диапазоне от 192 до 223.

3. Если символ является заглавной буквой, то его код уменьшается на 32, чтобы получить соответствующую строчную букву (код строчной буквы на 32 больше, чем заглавной в кодировке Windows-1251).

4. Полученный символ (строчная буква) выводится.

5. Если символ не принадлежит к заглавным буквам, выводится сообщение, что символ не является заглавной буквой.

***Описание алгоритма, используя псевдокод:***

НАЧАЛО

Ввод символа (char)

код\_символа = получить код символа

ЕСЛИ (код\_символа >= 192) И (код\_символа <= 223) ТО:

код\_строчного\_символа = код\_символа + 32

строчный\_символ = преобразовать код\_строчного\_символа в символ

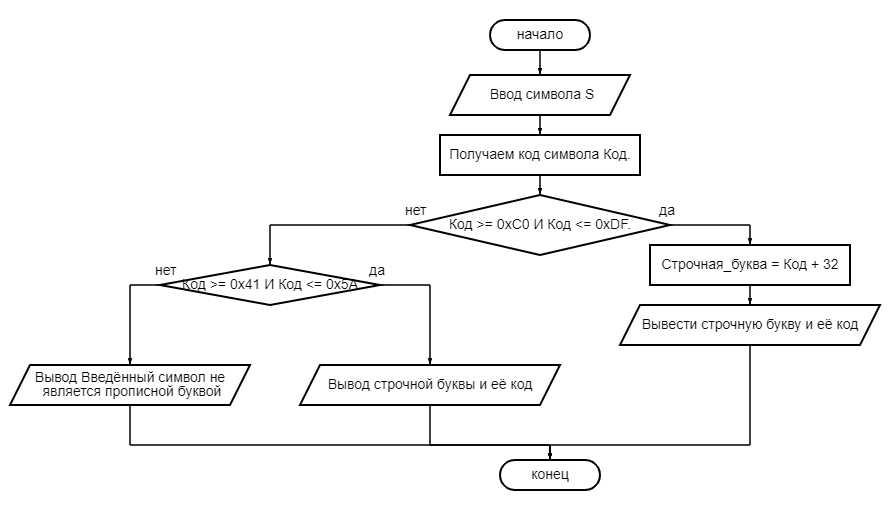
Вывести "Строчная версия символа: " + строчный\_символ

ИНАЧЕ:

Вывести "Символ не является заглавной буквой!"

КОНЕЦ

***Блок-схема алгоритма***



**№6**

*Вариант 7*

***Cловесно-формульное описание алгоритма:***

1. Начало алгоритма.

2. Ввод чисел:

* Пользователь вводит три вещественных числа.
* Обозначаем их как a, b и c.

3. Инициализация переменной для минимального значения:

* Инициализируем переменную min, присвоив ей значение первого числа: min= a.

4. Сравнение первого числа со вторым:

* Если второе число меньше текущего значения min, то присваиваем min значение второго числа:
  + Если (b < min) то min = b.

5. Сравнение текущего минимального числа с третьим:

* Если третье число меньше текущего значения min, то обновляем значение переменной min:
  + Если (c < min) то min = c.

6. Вывод результата:

* Выводим на экран значение переменной min, которая теперь содержит наименьшее число из трёх.

7. Конец алгоритма.

***Описание алгоритма, используя псевдокод:***

НАЧАЛО

Инициализация переменных a,b,c,min

Ввод a,b,c

ПРИСВОИТЬ min =a

ЕСЛИ b<min ТО min = b;

ИНАЧЕ continue

ЕСЛИ c<minТО min = b;

ИНАЧЕ continue

ВЫВОД минимум = min

КОНЕЦ

***Блок-схема алгоритма***

