Учреждение образования

Белорусский государственный технологический университет

Отчет по лабораторной №6

Факультет Информационных технологий

Программная инженерия

Чистова Анастасия Александровна

1курс, 7 группа, 1 подгруппа

Основы программной инженерии

Преподаватель: Наркевич Аделина Сергеевна

Минск, 2024

Задача:

Написать диалоговую программу, которая в зависимости от выбранного варианта использования выполняет действия пунктов 1, 2, 3, 4 для любого введенного с клавиатуры символа в стиле модульного программирования.

Входные данные: номер варианта использования программы, количество символов, символ.

Выходные данные: символы букв в строчном и прописном вариантах, код символов в кодировках, разница значений кодов.

Начало

нет

да

нет

нет

нет

да

да

да

да

Конец

Ошибка

Верхний регистр, upper, нижний регистр, lower, lower-upper

lower= код нижнего регистра

upper= код верхнего регистра

x >= 'A' and x <= 'Z' || x >= 'a' and x <= 'z'

Введите латинскую букву

1

2

3

4

1

Конец

4

Case 2

Case 3

Case 4

3

2

Case 1

1

Выберите вариант: 1,2,3,4

**Модуль 1**

Назначение: вычислить разницы кодов латинской буквы.

Входные данные: символ (латинская буква)

Выходные данные: Буква в верхнем и нижнем регистрах, код в ASCII, разница кодов

**Модуль 2**

Назначение: вычислить разницы кодов русской буквы.

Входные данные: символ (русская буква)

Выходные данные: Буква в верхнем и нижнем регистрах, код в Windows-1251 , разница кодов

2

Введите русскую букву

нет

Ошибка

x >= 'A' and x <= 'Я' || x >= 'a' and x <= 'я'

да

lower= код нижнего регистра

upper= код верхнего регистра

Верхний регистр, upper, нижний регистр, lower, lower-upper

Конец

**Модуль 3**

Назначение: вывести код цифры.

Входные данные: символ (цифра)

Выходные данные: Цифра, код в ASCII.

3

Введите цифру

Ошибка

нет

x >= '0' and x <= '9' |

да

Цифра, код

Конец

**Модуль 4**

Назначение: выход из программы.

Входные данные: символ.

Выходные данные: выход из программы.

4

Выход из программы

Конец

Определение разницы значений кодов русской буквы в прописном и строчном написании

Выход из программы

Вывод в консоль кода символа, соответствующего введенной цифре

Определение разницы значений кодов латинской буквы в прописном и строчном написании

Программа

Псевдокод:

**ПЕРВЫЙ ЭТАП** пошагового уточнения. Задается заголовок программы,

соответствующий ее основной функции:

**PROGRAM** Кодировка \_ разница;

**ВТОРОЙ ЭТАП** пошагового уточнения. Определяются основные действия на выбор:

**PROGRAM** Кодировка \_разница;

Определение\_ разницы \_ кодов \_ в \_ASCII \_ буквы \_ в \_ прописном и \_строчном \_ написании \_ в \_ латинском \_ алфавите.

Определение \_ разницы \_ кодов \_ в \_ Windows-1251 \_ буквы \_ в \_

Прописном \_ и \_ строчном \_ написании \_ в \_ русском \_алфавите.

Выводи \_в \_консоль \_кода \_символа \_соответствующего\_ введенной \_цифре.

Выход \_из \_программы.

**ENDPROGRAM**.

**ТРЕТИЙ ЭТАП** пошагового уточнения. **Детализация** фрагмента:

**PROGRAM** Кодировка \_разница;

**WHILE**

**BEGIN**

Определить \_ входит \_ ли \_ в \_ диапазон(0x042<=x<=0x07A).

**END**

Присвоить \_ код

Вычислить \_ разницу

Вывести \_ разница

**ENDPROGRAM**.

**ЧЕТВЁРТЫЙ ЭТАП** пошагового уточнения. **Детализация** фрагмента:

**PROGRAM** Кодировка \_разница;

**WHILE**

**BEGIN**

Определить \_ входит \_ ли \_ в \_ диапазон(0x41<=x<=0х44F)

**END**

Присвоить \_ код

Вычислить \_ разницу

Вывести \_ разница

**ENDPROGRAM**.

**ПЯТЫЙ ЭТАП** пошагового уточнения. **Детализация** фрагмента:

**PROGRAM** Кодировка \_разница;

**WHILE**

**BEGIN**

Определить \_ входит \_ ли \_ в \_ диапазон ( x>= 0 and x <= 9)

**END**

Присвоить \_ 0

Вычислить \_ код

Вывести \_ код

**ENDPROGRAM**.

**ШЕСТОЙ ЭТАП** пошагового уточнения. **Детализация** фрагмента:

**PROGRAM** Кодировка \_разница

**BEGIN**

Выход \_ из \_ программы

Иначе \_ошибка

**ENDPROGRAM**.

1. Парадигма программирования – это совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию).
2. Методология включает в себя модель вычислений для данного стиля. Методология разработки программного обеспечения – совокупность методов, применяемых на различных стадиях жизненного цикла программного обеспечения.
3. Императивное программирование (от греч. imper – действие) предполагает, что программа явно описывает алгоритм решения конкретной задачи (действия исполнителя), т.е. описывает как решать поставленную задачу. Декларативное программирование (лат. declaratio – объявление) – это предварительная реализация «решателя» для целого класса задач. Тогда для решения конкретной задачи этого класса достаточно декларировать в терминах данного языка только её условие: (исходные данные + необходимый вид результата) «Решатель» сам выполняет процесс получения результата, реализуя известный ему алгоритм решения.
4. Цели структурного программирования: − повысить надежность программ (улучшить структуру программы); − создание понятной, читаемой программы, которая исполняется, тестируется, конфигурируется, сопровождается и модифицируется без участия автора (создание ПП). Принципы разработки: − программирование «сверху-вниз» (нисходящее программирование); − модульное программирование с иерархическим упорядочением связей между модулями/подпрограммами «От общего к частному»
5. − следование; − ветвление; − цикл.
6. Технология структурного программирования базируется на следующих методах: − нисходящее проектирование (формализация алгоритма «сверху вниз»: движение от общего к частному); − пошаговое проектирование (нисходящая пошаговая детализация программы); − структурное проектирование (замена формулировки алгоритма на одну из синтаксических конструкций – последовательность, условие или цикл; программирование без goto); − одновременное проектирование алгоритма и данных (процессы детализации алгоритма и введение данных, необходимых для работы, идут параллельно); − модульное проектирование; − модульное, нисходящее, пошаговое тестирование.
7. «Разделяй и властвуй» - латинская формулировка принципа, лежащего в основе модульного проектирования.
8. Функциональная декомпозиция задачи – разбиение большой задачи на ряд более мелких, функционально самостоятельных подзадач – модулей.
9. Модуль, основные характеристики: − один вход и один выход (на вход программный модуль получает набор исходных данных, выполняет их обработку и возвращает набор выходных данных); − функциональная завершенность (модуль выполняет набор определенных операций для реализации каждой отдельной функции, достаточных для завершения начатой обработки данных); − логическая независимость (результат работы данного фрагмента программы не зависит от работы других модулей); − слабые информационные связи с другими программными модулями (обмен информацией между отдельными модулями должен быть минимален); − размер и сложность программного элемента должна быть в разумных рамках.
10. Технология модульного программирования базируется на следующих методах: − методы нисходящего проектирования (назначение – декомпозиция большой задачи на меньшие так, чтобы каждую подзадачу можно было рассматривать независимо.); − методы восходящего проектирования.

Дополнительное задание:

Задача:

Написать программу, которая в зависимости от введенного число выводила соответствующий день недели.

Входные данные: число

Выходные данные: соответствующий день недели

Конец

6

5

7

4

3

2

1

Case 7

Case 6

Case 5

Case 4

Case 3

Case 2

Case 1

7

6

5

4

3

2

1

Выберите вариант: 1,2,3,4,5,6,7

Начало

нет

нет

нет

нет

нет

нет

нет

да

да

да

да

да

да

да

**Модуль 4**

Назначение: выход Четверг.

Входные данные: цифра.

Выходные данные: Четверг.

**Модуль 3**

Назначение: выход Среда.

Входные данные: цифра.

Выходные данные: Среда.

**Модуль 2**

Назначение: выход Вторник.

Входные данные: цифра.

Выходные данные: Вторник.

**Модуль 1**

Назначение: выход Понедельник.

Входные данные: цифра.

Выходные данные: Понедельник.

1

Понедельник

Конец

Четверг

4

Конец

Конец

Среда

3

Конец

Вторник

2

**Модуль 7**

Назначение: выход Воскресенье.

Входные данные: цифра.

Выходные данные: Воскресенье.

**Модуль 6**

Назначение: выход Суббота.

Входные данные: цифра.

Выходные данные: Суббота.

**Модуль 5**

Назначение: выход Пятница.

Входные данные: цифра.

Выходные данные: Пятница.

Конец

Воскресенье

7

6

Конец

Суббота

Конец

5

Пятница

Выберите вариант

Вывести воскресенье

Вывести Суббота

Вывести Пятница

Вывести Четверг

Вывести Среда

Вывести Вторник

Вывести Понедельник

5

7

6

4

3

2

1