Высшее учебное заведение

“Белорусский Государственный Технологический Университет”

Факультет информационных технологий

Специальность программная инженерия

1 курс, 7 группа, 1 подгруппа

Основы программной инженерии

Лабораторная работа №6

Тарасевич Екатерина Александровна

Задание №1

В программе дополнительно предусмотрены возможности ввода с клавиатуры нескольких символов последовательно

Задание №2

Написать диалоговую программу, которая в зависимости от выбранного варианта 1, 2, 3 или 4 будет выполнять следующие действия:

1. Определяет разницу значений кодов в ASCII буквы разного регистра, если введен символ латинского алфавита, иначе вывод сообщения об ошибке.
2. Определяет разницу значений кодов в Windows-1251 буквы разного регистра, если введен символ русского алфавита, иначе вывод сообщения об ошибке.
3. Вывод в консоль кода символа, соответствующего введенной цифре, иначе вывод сообщения об ошибке.
4. Выход из программы

В случае введения нескольких символов подряд, программа будет выводить сообщение для каждого символа в соответствии с выбранным вариантом, перечисленных выше.

Задание №3

Входные данные: введенные символы с клавиатуры

Выходные данные: сообщения с информацией о разницы кодов букв в разных регистрах, сообщения об ошибках, коды символов, если это цифры.

Задание №4



Задание №6

**Модуль letters**

Содержит функции(latinskie, russkie), которые определяют, является ли символ латинской или русской буквой, а также функции (AllLatinskie, AllRusskie), которые определяют, являются ли введенные символы латинскими буквами

Функция latinskie принимает символ, выводит коды символов в разном регистре и разницу кодов разного регистра буквы, если введенный символ является латинской буквой или сообщение об ошибке.

Функция russkie принимает символ, выводит коды символов в разном регистре и разницу кодов разного регистра буквы, если введенный символ является русской буквой или сообщение об ошибке.

Функция AllLatinskie прогоняет каждый символ через функцию latinskie

Функция AllRusskie прогоняет каждый символ через функцию russkie

Алгоритм модуля:

Функция latinskie

1. 1.если введенный символ является латинской буквой, то возвращаем сообщение с разницей кодов разного регистра буквы, иначе п.2
2. 2.выводит сообщение об ошибке

Функция russkie

1. 1.если введенный символ является русской буквой, то возвращаем сообщение с разницей кодов разного регистра буквы, иначе п.2
2. 2.возвращает сообщение об ошибке

Функция AllLatinskie

1. Вывод сообщения “Введите символ”
2. Ввод строки
3. Для каждого символа строки осуществляется переход к функции latinskie

Функция AllRusskie

1. Вывод сообщения “Введите символ”
2. Ввод строки
3. Для каждого символа строки осуществляется переход к функции russkie

**Модуль number**

Содержит функцию (number, numbers) которая определяет, является ли символ цифрой

Функция number принимает символ и если символ является цифрой, то выводит символ и его код, иначе выводит сообщение об ошибке.

Функция numbers прогоняет каждый символ через функцию number

Алгоритм модуля:

1. вывод сообщения “введите символ”
2. ввод символа
3. Для каждого символа строки осуществляется переход к функции number





Задание №8

***Первый этап*** пошагового уточнения. **Задается** заголовок программы, соответствующий ее основной функции:

**Рrogram** Обработка\_символов

***Второй этап*** пошагового уточнения. **Определяются** основные действия:

**Program** Обработка\_символов;

Выбрать\_вариант\_обработки;

Получить\_символ;

Обработать\_полученный\_символ;

Вывести\_результат\_обработки;

**EndРrogram**

***Третий этап*** пошагового уточнения. **Детализация** фрагмента:

**Program** Обработка\_символов;

**Do**

Выбрать\_вариант\_обработки;

Вариант\_1: обработка\_латинских\_букв;

Вариант\_2: обработка\_русских\_букв;

Вариант\_3: обработка\_цифр;

Вариант\_4: выход;

**While** (вариант\_обработки\_не\_равен\_выходу);

**EndРrogram**.

***Четвертый этап*** пошагового уточнения. **Детализация** фрагмента:

**Вариант\_1**;

ВВОДсимволов

ЦИКЛ(для каждого символа)

ЕСЛИ символ - латинская буква

ДА ВЫВОД сообщение с разницей значений кодов разного регистра буквы

НЕТ ВЫВОД сообщения об ошибке

КОНЕЦ ЦИКЛА

**Вариант\_2**;

ВВОДсимволов

ЦИКЛ(для каждого символа)

ЕСЛИ символ - русская буква

ДА ВЫВОД сообщение с разницей значений кодов разного регистра буквы

НЕТ ВЫВОД сообщения об ошибке

КОНЕЦ ЦИКЛА

**Вариант\_3**;

ВВОДсимволов

ЦИКЛ(для каждого символа)

ЕСЛИ символ - цифра

ДА ВЫВОД символа и его кода

НЕТ ВЫВОД сообщения об ошибке

КОНЕЦ ЦИКЛА

Ответы на вопросы

1. Парадигма программирования-это совокупность идей и понятий, определяющих стиль написания компьютерных программ(подход к программированию)

- Императивная парадигма

- Декларативная парадигма

- Объектно-ориентированная парадигма

- Функциональная парадигма

- Логическая парадигма

2. Методология разработки программного обеспечения-это совокупность принципов, практик и процессов, направленных на создание качественного программного продукта. Методология определяет как правильно и эффективно вести проект от начала и до конца.

- Управление рисками

- Повышение качества продукта

- Повышение производительности разработки

- Соблюдение сроков и бюджета

3.Отличие императивного программирования от декларативного:

В императивном программировании программист описывает последовательность шагов для решения задачи, в то время как в декларативном — описывает требуемые результаты, не заботясь о том, как они будут достигнуты.

4. Цели и принципы структурного программирования:

- Цели: улучшение читаемости, понимания и сопровождения программ; минимизация ошибок; улучшение тестируемости.

- Основные принципы:

- Программа должна быть разделена на блоки или подпрограммы.

- Использование трёх структур управления: последовательность, выбор и повторение.

- Минимизация использования переходов.

5. Базовые конструкции языка программирования, лежащие в основе структурного программирования:

- Последовательность- выполнение инструкций одна за другой.

- Выбор (ветвление)- условное выполнение различных блоков кода.

- Повторение — выполнение кода несколько раз.

6. Методы структурного программирования:

-Декомпозиция задач- разделение программы на независимые, логически завершённые части.

- Использование подпрограмм-разбиение на мелкие блоки с отдельной функциональностью.

- Избежание использования безусловных переходов.

- Модульность и инкапсуляция-разбиение программы на модули и скрытие реализации.

7. Принцип модульного проектирования- разделение системы на независимые и взаимодействующие компоненты (модули), каждый из которых выполняет конкретную задачу и может быть разработан, протестирован и использован отдельно. Это повышает гибкость, повторное использование и улучшает поддержку системы.

8. Функциональная декомпозиция задачи- это процесс разбиения сложной задачи на более мелкие, взаимозависимые подзадачи, каждая из которых решается с помощью отдельной функции или процедуры. Такой подход позволяет упростить проектирование, тестирование и модификацию программы.

9. Основные характеристики модуля:

- Инкапсуляция: скрытие внутренней реализации и предоставление только необходимого интерфейса.

- Абстракция: представление функциональности модуля на высоком уровне, без необходимости вникать в детали.

- Независимость: возможность разработки, тестирования и модификации модуля без значительного воздействия на другие части системы.

- Повторное использование: модуль может быть использован в других частях программы или в других проектах.

- Интерфейс: чётко определённый набор операций и данных, которые доступны пользователю модуля.

10.Методы модульного программирования:

- Декомпозиция: разбиение системы на отдельные модули.

- Использование интерфейсов: определение чётких интерфейсов для взаимодействия между модулями.

- Инкапсуляция: скрытие внутреннего состояния модуля и управление доступом к данным через функции.

- Абстракция: создание модулей, которые решают конкретные задачи, не вникая в детали их реализации.

- Независимость: обеспечение того, чтобы изменение одного модуля не влияло на другие.