**ПРАКТИЧЕСКАЯ РАБОТА №5**

Тема: Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования.

Цель разработка алгоритма для создания программного продукта и анализ предметной области.

Ход работы:

Задание 1

Каковы этапы жизненного цикла информационных систем и их основное содержание?

* Определение потребностей и возможностей. На данном этапе происходит оценка текущих проблем и потребностей, а также выявление возможностей для улучшения бизнес-процессов.
* Анализ требований. Здесь проводится более подробное изучение требований пользователей, бизнес-процессов и технических ограничений. Разрабатываются спецификации и требования к новой информационной системе.
* Проектирование. На данном этапе создается концептуальное и техническое проектирование информационной системы, включая выбор архитектуры, технологий и разработку дизайна пользовательского интерфейса.
* Разработка. Здесь осуществляется написание программного кода, создание баз данных, тестирование и интеграция компонентов системы.
* Внедрение. На этом этапе информационная система становится доступной для использования. Проводится обучение пользователей, настраивается окружающая среда, и происходит перенос данных в новую систему.
* Эксплуатация и поддержка. После внедрения начинается фаза эксплуатации, включая техническую поддержку, обслуживание и улучшение системы в соответствии с изменяющимися потребностями бизнеса.
* Фаза вывода из эксплуатации. В конечном итоге информационная система может быть выведена из эксплуатации из-за устаревания, изменения бизнес-потребностей или других причин.

Задание 2

Как представляется функциональная модель деятельности в методологии IDEF0?

Методология IDEF0 представляет функциональную модель деятельности в виде диаграммы, называемой «блок-схемой функций». Принципиальная иерархическая структура диаграммы IDEF0 позволяет представлять деятельность на различных уровнях детализации, начиная от общего контекста и заканчивая более детальными описаниями функций.

Задание 3

Каковы основные объекты диаграмм функциональной модели по методологии IDEF0?

В методологии IDEF0 основными объектами диаграмм функциональной модели являются:

* Блок функций (Function Box). Блок функций представляет собой основную функцию или деятельность в системе. Каждый блок функций имеет уникальный идентификатор и содержит краткое описание выполняемой функции. Блоки функций используются для идентификации и описания ключевых функциональных элементов системы.
* Стрелки (Arrows). Стрелки на диаграмме указывают на поток информации или материалов между блоками функций. Они представляют собой связи между функциональными блоками и показывают направление потока данных или материалов в системе.
* Контекст (Context). Контекстное окружение может быть представлено отдельным блоком, который описывает общую систему и связи между ее внешней средой и внутренними функциями. Контекст помогает определить границы системы и ее взаимодействие с внешними факторами.
* Декомпозиция (Decomposition). Декомпозиция представляет собой процесс разбиения блока функций на более детальные уровни с использованием дополнительных блоков функций. Это позволяет более подробно описывать внутренние процессы и взаимосвязи между функциями.
* Механизмы управления (Control Mechanisms). Дополнительные символы, такие как стрелки управления и различные метки, могут использоваться для обозначения механизмов управления в системе. Это включает в себя принятие решений, управление потоком данных и другие аспекты управления процессами.

Задание 4

Что обозначают работы в диаграммах функциональной модели, как они отображаются по методологии IDEF0?

Работы в диаграммах функциональной модели по методологии IDEF0 обозначаются буквенно-цифровыми обозначениями (A0, A1, A2 и т. д.), которые указывают на порядок выполнения функций внутри блоков. Например, A0 может быть основной функцией, A1 - дополнительной функцией, выполняемой внутри A0, и так далее.

Задание 5

Для чего предназначены стрелки в диаграммах функциональной модели, каковы их типы и виды?

Стрелки в диаграммах функциональной модели предназначены для обозначения потоков данных или управления между функциональными блоками. Основные типы и виды стрелок:

* Стрелки данных (Data Arrows): обозначают поток данных между функциональными блоками.
* Стрелки управления (Control Arrows): обозначают поток управления между функциональными блоками.
* Стрелки обратной связи (Feedback Arrows): обозначают обратные связи или циклы в процессе.

Задание 6

Для чего предназначен словарь стрелок?

Словарь стрелок предназначен для объяснения значений и типов стрелок, используемых в диаграммах функциональной модели. В словаре стрелок указывается, какие данные передаются, какой тип потока (например, управления или данных) используется, и как они взаимодействуют между функциональными блоками.

Классификация средств отладки прикладного ПО для встраиваемых микропроцессоров может включать следующие категории:

Эмуляторы: устройства, которые имитируют работу целевого устройства или микропроцессора на компьютере разработчика.

Интегрированные среды разработки (IDE): программные пакеты, которые объединяют в себе средства разработки и отладки, такие как средства мониторинга состояния программы, анализа кода и т. д.

Инструменты для анализа времени выполнения: позволяют отслеживать время выполнения программы, идентифицировать узкие места и оптимизировать код.

Аппаратные отладчики: физические устройства, подключаемые к целевой системе для мониторинга и управления ее работой.

Виды и особенности аппаратных средств отладки ПО могут включать:

Эмуляторы: обеспечивают точное моделирование работы целевого устройства, позволяя программистам отлаживать код на компьютере без реального оборудования.

Интерфейсы JTAG: используются для мониторинга и отладки встроенных систем, позволяют читать и записывать данные в память микроконтроллера, управлять его работой и тестировать периферийные устройства.

Отладочные платы: специализированные платы, которые предоставляют доступ к различным интерфейсам и периферийным устройствам для тестирования и отладки.

Основные функции программных средств отладки ПО включают:

Мониторинг состояния программы: отслеживание значений переменных, стека вызовов, состояния регистров и т. д.

Управление выполнением программы: возможность пошагового выполнения, установки точек останова, изменения значения переменных и т. д.

Анализ кода: возможность просмотра исходного кода, поиска ошибок, оценки производительности и т. д.

Тестирование: автоматизация процесса тестирования программы, создание и выполнение тестовых сценариев.

Пошаговое выполнение программы позволяет программисту выполнить код программы пошагово, следить за изменением переменных, контролировать выполнение функций и т. д. Основные возможности включают в себя выполнение пошагово по инструкциям, по функциям или по блокам кода, установку точек останова, анализ состояния программы на каждом шаге выполнения.

Прогон программы с контрольными точками позволяет программисту установить точки останова в коде программы, где выполнение программы автоматически останавливается для анализа состояния программы. Это позволяет эффективно идентифицировать проблемные участки кода и отлаживать программу.

Контрольные точки могут быть разных типов:

Обычные точки останова: программа останавливается при достижении определенной строки кода или функции.

Условные точки останова: программа останавливается только в случае выполнения определенного условия.

Ловушки: используются для отслеживания определенных событий или исключений в программе.

Счетчики: позволяют установить ограничение на количество выполнений определенного участка кода.