**Отчет по лабораторной работе №6**

**Тема: Разработка модульной структуры проекта (диаграммы модулей)**

**Цель работы**

Изучение процесса разработки модульной структуры программного обеспечения с использованием структурных карт Константайна.

**Оборудование и программное обеспечение**

* Персональный компьютер.
* Среда программирования: Visual Studio 2019.
* Инструмент для создания структурных карт: EasyCASE Professional Version 4.21.016.

**Ход работы**

1. **Разработка иерархической диаграммы модулей**  
   В соответствии с заданием, модульная структура подсистемы обслуживания клиента по кредитной карте спроектирована как дерево, состоящее из следующих модулей:
   * **Головной модуль (Main Module)**:  
     Организует общее управление подсистемой.  
     Вызов дочерних модулей осуществляется в следующем порядке:
     1. Модуль управления устройством считывания карты.
     2. Модуль аутентификации.
     3. Модуль обработки запросов.
   * **Модуль управления устройством считывания кредитной карты (Credit Cart Control Module)**:  
     Выполняет обработку карты (ввод, считывание данных, удаление).
   * **Модуль аутентификации (Authentication Module)**:  
     Получает от клиента пароль, проверяет его корректность и возвращает результат аутентификации.
   * **Модуль получения и обработки запросов (Reception and Processing Module)**:  
     Выполняет обслуживание клиента, включая:
     1. Проверку возможности выполнения запроса.
     2. Обработку банковской документации.
     3. Распечатку баланса или операций.
     4. Выдачу наличных.
   * **Модуль данных кредитной карты (Credit Cart Data Module)**:  
     Управляет данными клиента: паролем, атрибутами клиента и лимитом средств.

**Диаграмма модулей:**  
*(схема, соответствующая описанным отношениям, может быть нарисована в EasyCASE или другой схожей программе)*

mathematica

Копировать код

Main Module

├── Credit Cart Control Module

├── Authentication Module

└── Reception and Processing Module

└── Credit Cart Data Module

1. **Спецификация модулей**  
   Каждому модулю была разработана спецификация, включающая:
   * **Синтаксическую спецификацию**:  
     Описание входных и выходных параметров, их типов, формата передачи.
   * **Функциональную спецификацию**:  
     Определение семантики функций, выполняемых модулем.

**Пример спецификаций:**

**1. Main Module**

* **Синтаксис входов**:
  + Init(): инициализация.
  + CallModule(name: string): вызов подчиненного модуля.
* **Выходы**:
  + status: bool: результат успешного завершения выполнения.
* **Функциональность**:
  + Управляет последовательностью вызовов дочерних модулей.

**2. Authentication Module**

* **Синтаксис входов**:
  + Authenticate(parol: string) -> bool: принимает пароль.
* **Выходы**:
  + result: bool: результат проверки пароля.
* **Функциональность**:
  + Проверяет подлинность клиента.

1. **Программирование модулей**  
   В ходе лабораторной работы реализована нисходящая разработка.
   * **Шаг 1**: Разработка головного модуля.
   * **Шаг 2**: Реализация и тестирование дочерних модулей с использованием имитаторов.
   * **Шаг 3**: Интеграция всех модулей в единое приложение.

**Вывод**

В результате работы была разработана модульная структура подсистемы обслуживания клиента банкомата по его кредитной карте. Использование структурных карт Константайна позволило создать иерархию модулей с минимальным сцеплением и максимальной связностью. Применение нисходящей разработки обеспечило целостность архитектуры.

**Контрольные вопросы**

1. **Цель разработки модульной структуры**:  
   Упрощение проектирования, реализации и тестирования ПО за счет деления на независимые модули.
2. **Понятие программного модуля**:  
   Логически обособленная часть программы, выполняющая определенную функцию.
   * **Передача управления**: вызов функции или метода модуля.
   * **Связь по данным**: передача и прием параметров или доступ к общей памяти.
3. **Виды связности модулей**:
   * Управляющая.
   * Данные.
   * Процедурная.
4. **Виды целостности модулей**:
   * Информационная.
   * Функциональная.
5. **Типовые модульные структуры**:
   * Древовидная.
   * Сросшаяся (с перекрестными связями).
6. **Проектирование модульной структуры с помощью структурных карт**:  
   Позволяет визуализировать иерархию и связи между модулями системы.
7. **Построение структурных карт с помощью EasyCASE**:
   * Создание нового проекта.
   * Определение модулей и их взаимосвязей.
   * Генерация схемы.