**Лабораторная работа № 5. Разработка модульной структуры проекта (диаграммы модулей)**

**Цель работы**

Изучить процесс проектирования модульной структуры программного обеспечения с использованием структурных карт Константайна для обеспечения высокой прочности модулей и минимального сцепления между ними.

**Ход работы**

**1. Анализ проектируемой системы**

Проектируем подсистему обслуживания клиента по его кредитной карте в банкомате.  
Основные модули:

1. **Головной модуль (Main module)**
   * Организует общий процесс управления системой.
2. **Модуль управления устройством считывания карты (Credit Card Control Module)**
   * Обеспечивает ввод, чтение и удаление карты.
3. **Модуль аутентификации (Authentication Module)**
   * Обрабатывает ввод данных для проверки подлинности клиента.
4. **Модуль получения и обработки запроса на обслуживание (Reception and Processing Module)**
   * Выполняет обслуживание клиента: запросы баланса, выдачу средств, распечатку квитанций и т. д.
5. **Модуль данных карты (Credit Card Data Module)**
   * Содержит информацию о клиенте, пароле и лимите на счету.

**2. Построение структурной карты**

На основе задания построим структурную карту, показывающую связи между модулями (Рис. 1).

**Описание связей модулей:**

* **Головной модуль** управляет всеми остальными модулями.
  + Вызывает модуль управления картой для чтения/удаления карты.
  + Передает управление модулю аутентификации для проверки клиента.
  + Передает управление модулю обработки запроса, если аутентификация прошла успешно.
* **Модуль управления устройством считывания карты**:
  + Взаимодействует с клиентом для ввода/удаления карты.
  + Передает данные о карте в общую область памяти.
* **Модуль аутентификации**:
  + Использует данные из памяти для проверки пароля.
  + Возвращает результат аутентификации в головной модуль.
* **Модуль обработки запроса**:
  + Работает с данными клиента (атрибуты и лимит).
  + Выполняет запросы клиента, такие как вывод наличных и распечатка.

**3. Разработка спецификации модулей**

**Головной модуль (Main module)**

* **Входы:**
  + Событие от клиента.
* **Выходы:**
  + Вызов подчиненных модулей (обработка карты, аутентификация, запросы обслуживания).
* **Функциональность:**
  + Принимает запрос клиента.
  + Вызывает модуль управления картой для считывания данных.
  + Вызывает модуль аутентификации.
  + В зависимости от результата:
    - При успешной аутентификации вызывает модуль обработки запросов.
    - В случае ошибки — удаляет карту.

**Модуль управления устройством считывания карты (Credit Card Control Module)**

* **Входы:**
  + Событие: вставка/удаление карты.
* **Выходы:**
  + Данные карты (передаются в память).
* **Функциональность:**
  + Инициирует процесс считывания карты.
  + Сохраняет данные карты в общую область памяти.
  + При запросе удаляет карту.

**Модуль аутентификации (Authentication Module)**

* **Входы:**
  + Данные карты (пароль).
* **Выходы:**
  + Результат проверки (Authentication flag).
* **Функциональность:**
  + Сравнивает введенный клиентом пароль с данными карты.
  + Возвращает результат аутентификации.

**Модуль обработки запросов (Reception and Processing Module)**

* **Входы:**
  + Данные клиента (атрибуты, лимит).
  + Запрос на обслуживание.
* **Выходы:**
  + Результат выполнения запроса (распечатка квитанции, выдача наличных).
* **Функциональность:**
  + Проверяет возможность выполнения запроса.
  + Выполняет запрос.
  + Обновляет данные клиента.

**Модуль данных карты (Credit Card Data Module)**

* **Функциональность:**
  + Сохраняет данные карты (пароль, лимит).

**4. Диаграмма модульной структуры**

На основе анализа модулей и их взаимодействия построим диаграмму. Диаграмма строится в программе EasyCASE.

**5. Программная реализация**

Создание заглушек для модулей в Visual Studio для тестирования системы.

Пример заглушки модуля аутентификации:

cpp

Копировать код

#include <iostream>

#include <string>

using namespace std;

bool Authenticate(string enteredPassword, string storedPassword) {

if (enteredPassword == storedPassword) {

cout << "Authentication successful." << endl;

return true;

}

cout << "Authentication failed." << endl;

return false;

}

int main() {

string storedPassword = "1234"; // Заглушка данных карты

string enteredPassword;

cout << "Enter password: ";

cin >> enteredPassword;

bool result = Authenticate(enteredPassword, storedPassword);

return result ? 0 : 1;

}

**Вывод**

В ходе работы изучен процесс разработки модульной структуры на основе структурных карт Константайна. Построены спецификации модулей, а также диаграмма модульной структуры системы. Разработана заглушка модуля для тестирования.

**Ответы на контрольные вопросы**

1. **Цель разработки модульной структуры**:
   * Обеспечение высокой прочности и минимального сцепления модулей.
2. **Понятие программного модуля**:
   * Независимый блок программы, выполняющий определенные функции. Передача управления осуществляется через вызов функций или методов.
3. **Виды связности модулей**:
   * Данные, управление, общий ресурс.
4. **Виды целостности модулей**:
   * Логическая, временная, процессная.
5. **Типовые модульные структуры**:
   * Иерархическая, линейная.
6. **Проектирование структурных карт**:
   * Использование дерева подчиненности модулей.
7. **Построение структурных карт**:
   * С помощью EasyCASE.