|  |  |  |
| --- | --- | --- |
|  | **Федеральное государственное бюджетное образовательное**  **учреждение высшего образования**  **«Пензенский государственный университет»**  **(ФГБОУ ВО «Пензенский государственный университет»)**  **Нижнеломовский филиал ФГБОУВО**  **«Пензенский государственный университет»**  **(НлФ ФГБОУ ВО «ПГУ»)** |  |

**Практическая работа №4**

**по дисциплине:** «Системное программирование»

**на тему:** «Создание и использование динамических библиотек»

Выполнили:

студент группы 23НФПО

Кириллов А.О.

Веденяпин Д.А.

Мосман А.П.

Принял:

Преподаватель

Хлынцев А.А.

Нижний Ломов, 2025 г.

**Ход работы:**

1. Создали динамически подключаемую библиотеку (DLL – Dynamic Link Library), которая реализует функцию сложения (функция принимает два параметра «a», «b» и возвращает результат их суммы), вычитание (функция принимает два параметра «a», «b» и возвращает результат их разности), умножения (функция принимает два параметра «a», «b» и возвращает результат их произведения), деления (функция принимает два параметра «a», «b» и возвращает результат их частного) (рисунок 1-4).

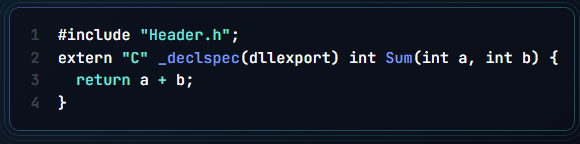
****

Рисунок 1 – Код функции сложения

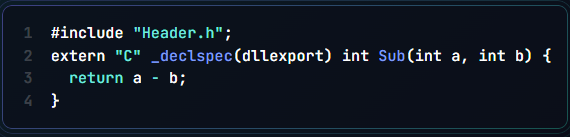


Рисунок 2 – Код функции вычитания

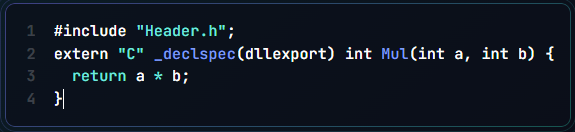


Рисунок 3 – Код функции умножения

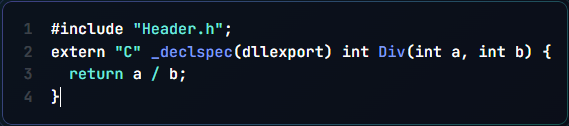


Рисунок 4 - Код функции деления

1. Создали заголовочный файл проекта «Header.h» (рисунок 5).

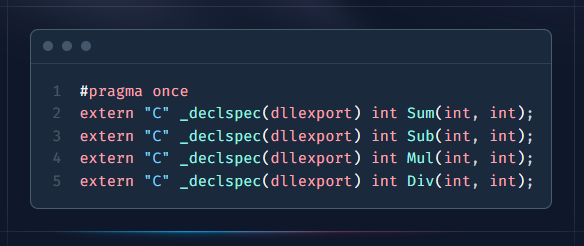


Рисунок 5 – Код заголовочного файла

1. Создали главный исполняемый файл проекта для использования DLL, код которого загружает DLL библиотеку и вызывает из нее четыре математические функции (сложение, вычитание, умножение и деление), затем выводит результаты этих операций в консоль (рисунок 6-7).

****

Рисунок 6 – Код главного исполняемого файла

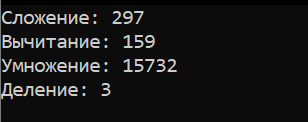


Рисунок 7 – Результат работы программ

**Вывод:** научились создавать и использовать DLL (динамические библиотеки) — это важный и неотъемлемый элемент архитектуры программного обеспечения в операционной системе Windows, который обеспечивает модульность, повторное использование кода и динамическое связывание компонентов приложений. Они позволяют разработчикам создавать универсальные и расширяемые программные решения, разделяя функциональность на отдельные компоненты, которые могут быть загружены и использованы по мере необходимости, что значительно повышает эффективность использования ресурсов системы и ускоряет процесс разработки. Одним из главных преимуществ DLL является возможность обновления или расширения функциональности без необходимости перекомпиляции и повторной сборки всей программы: достаточно заменить или добавить новую DLL, что делает системы более гибкими и удобными в обслуживании. Это особенно актуально для крупных программных платформ и систем с высокой степенью модульности, таких как операционные системы, серверные приложения, игровые движки и корпоративные системы, где требования к расширяемости и обновляемости критичны. DLL содержат не только функции и классы, но также ресурсы, такие как изображения, строки, диалоги и другие элементы интерфейса, что позволяет централизованно управлять ресурсами и снижает их дублирование. Важной особенностью является возможность совместного использования DLL несколькими приложениями одновременно, что способствует экономии системных ресурсов, снижению затрат на разработку и поддержку, а также повышению стабильности системы за счет уменьшения дублирования кода и устранения ошибок в отдельных компонентах. В процессе работы системы, при запуске программы или при необходимости вызова определенной функции, операционная система динамически загружает соответствующие DLL, связывая вызовы функций с их реализациями, что обеспечивает гибкое управление зависимостями и повышает производительность за счет ленивой или предварительной загрузки библиотек. В свою очередь, это требует аккуратного управления версиями DLL, чтобы избежать конфликтов и ошибок, связанных с несовместимостью, так как наличие нескольких версий одной библиотеки в системе может привести к трудно диагностируемым сбоям. Поэтому внедрение механизмов контроля версий, цифровой подписи и автоматического обновления является важной частью разработки надежных и безопасных приложений, использующих DLL. Также важно учитывать вопросы безопасности: злоумышленники могут использовать уязвимости DLL для внедрения вредоносных библиотек или проведения DLL-инъекций, что требует внедрения дополнительных мер защиты, таких как контроль загрузки, проверка подписи и использование механизмов защиты памяти. В современных приложениях активно применяются технологии автоматической сборки, тестирования и обновления DLL, а также механизмы цифровой подписи для повышения доверия и безопасности. Кроме того, в контексте разработки используется понятие COM (Component Object Model), которое позволяет создавать сложные компоненты и управлять ими через интерфейсы, что еще больше расширяет возможности использования DLL. В целом, DLL являются мощным и универсальным инструментом для разработки программного обеспечения, обеспечивающим высокий уровень модульности, расширяемости, повторного использования и безопасности, что делает их незаменимыми в современных системах и приложениях различной сложности и назначения.