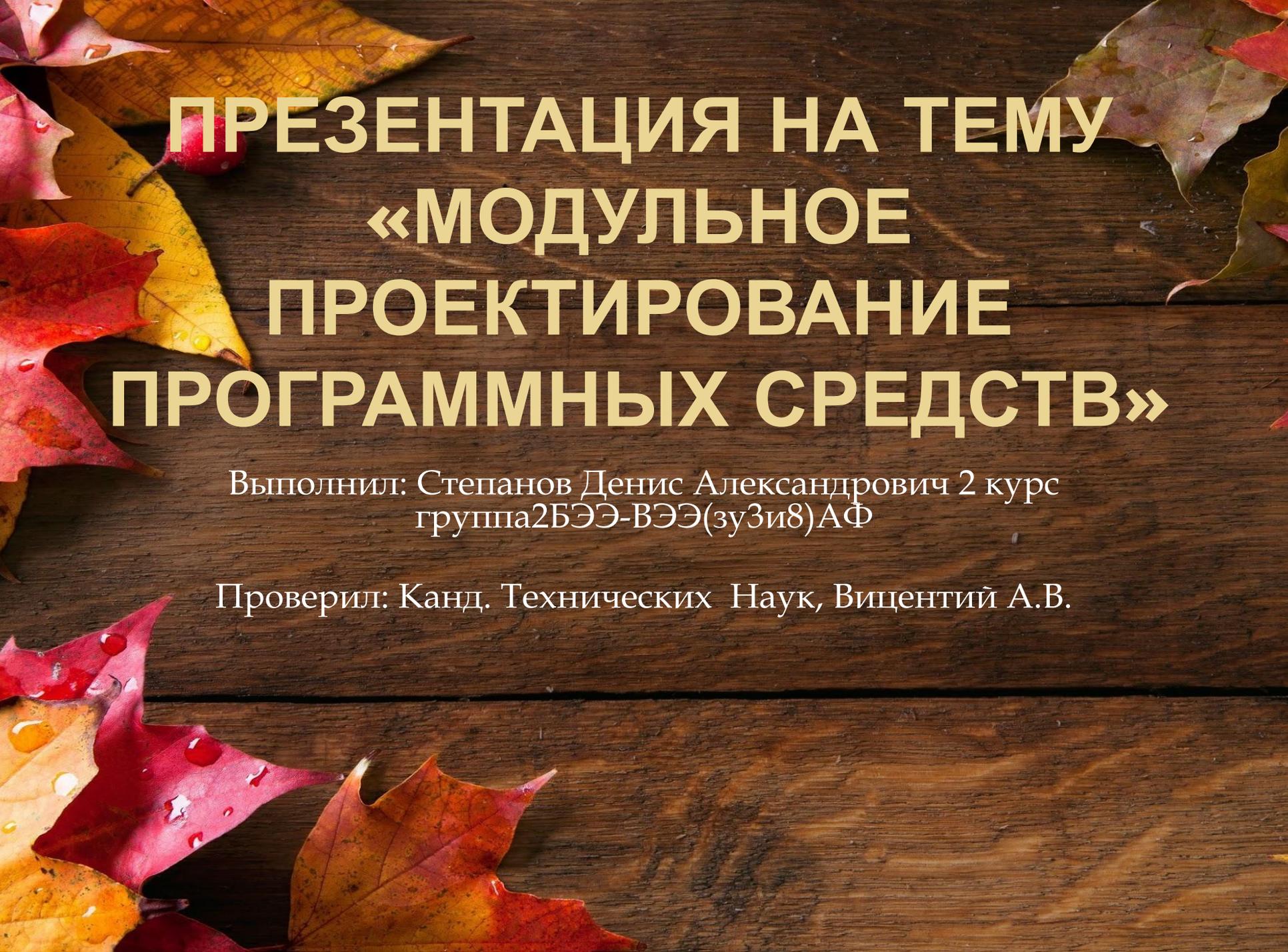
The background of the slide features a dark, textured wooden surface. Scattered around the edges are several autumn leaves in various shades of red, orange, and yellow. Some leaves have small, clear water droplets on them, adding a fresh, natural feel to the design.

**ПРЕЗЕНТАЦИЯ НА ТЕМУ
«МОДУЛЬНОЕ
ПРОЕКТИРОВАНИЕ
ПРОГРАММНЫХ СРЕДСТВ»**

The background features a dark, textured wooden surface with several autumn leaves scattered across it. The leaves are in various stages of decay, showing shades of yellow, orange, and red. Some leaves have small water droplets on them, suggesting a recent rain. The overall composition is warm and seasonal.

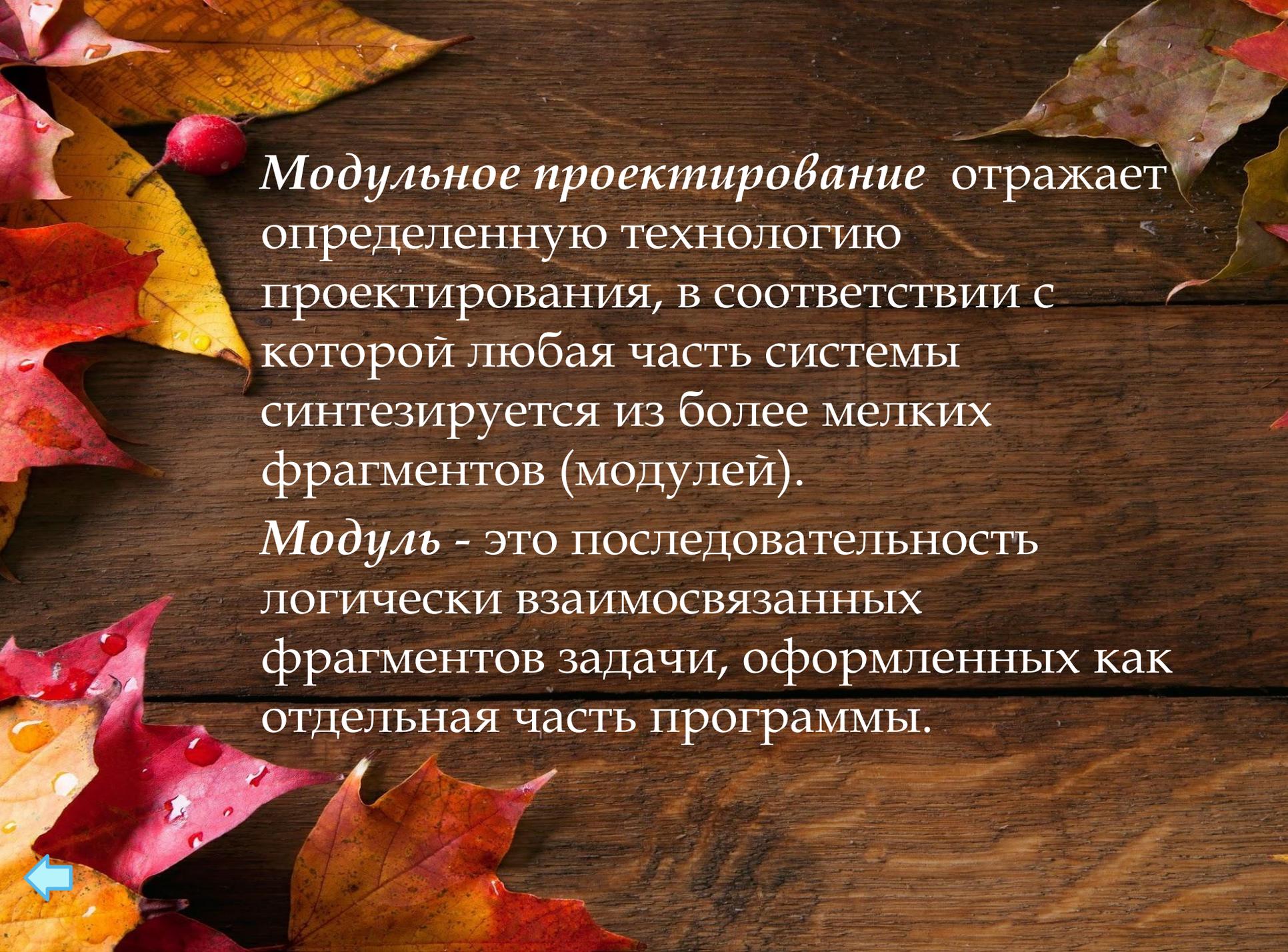
ПРЕЗЕНТАЦИЯ НА ТЕМУ «МОДУЛЬНОЕ ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ»

Выполнил: Степанов Денис Александрович 2 курс
группа 2БЭЭ-ВЭЭ(зу3и8)АФ

Проверил: Канд. Технические Наук, Вицентий А.В.

Содержание

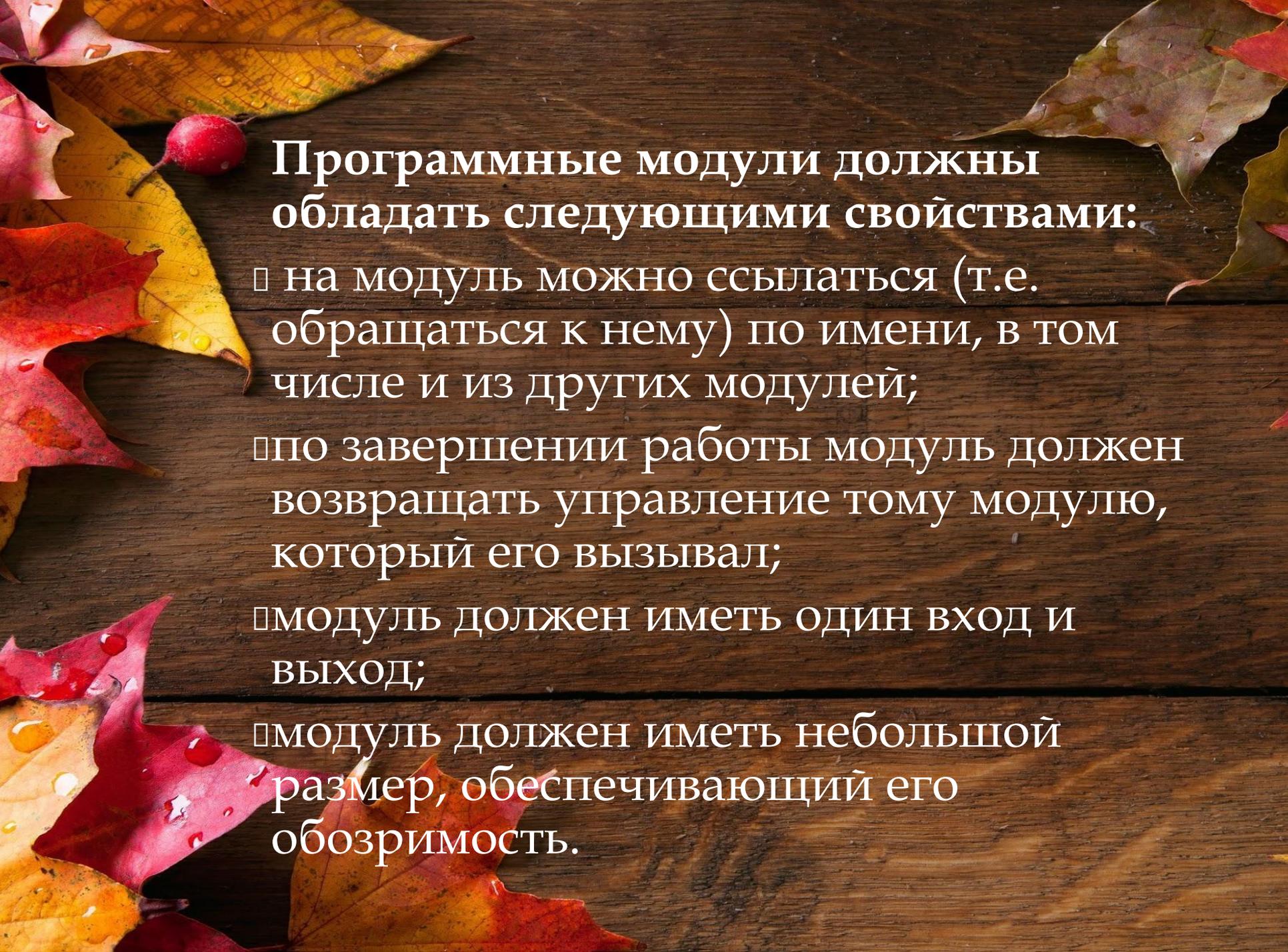
- ▣ Понятие модульного проектирования и модуля
- ▣ Свойства программных модулей
- ▣ Преимущества модульного принципа
- ▣ Недостатки модульного принципа
- ▣ Понятие модульной программы и принципы модульности программы
- ▣ Методы проектирования модульных программ
- ▣ Краткое описание методов проектирования модульных программ



Модульное проектирование отражает определенную технологию проектирования, в соответствии с которой любая часть системы синтезируется из более мелких фрагментов (модулей).

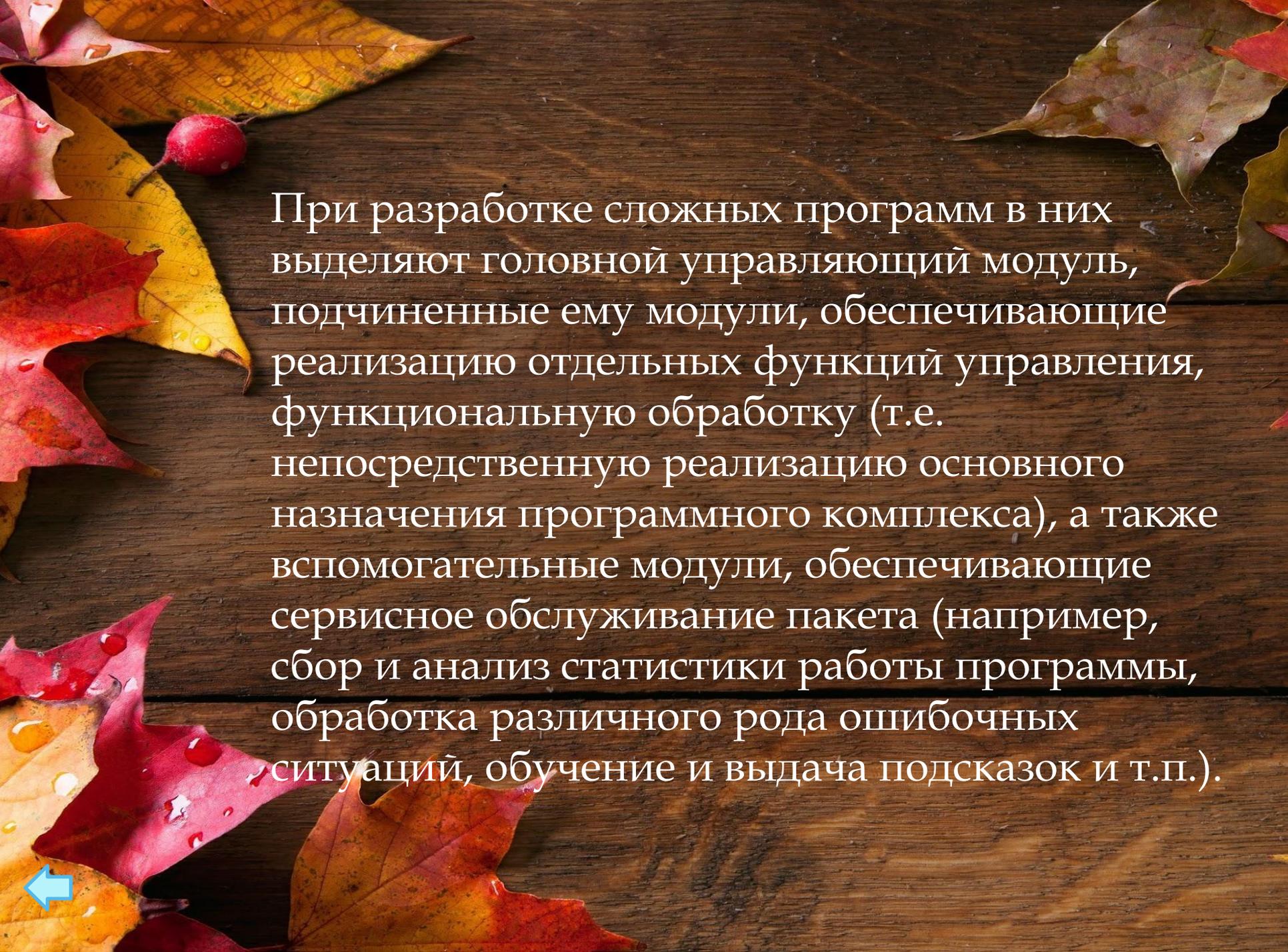
Модуль - это последовательность логически взаимосвязанных фрагментов задачи, оформленных как отдельная часть программы.





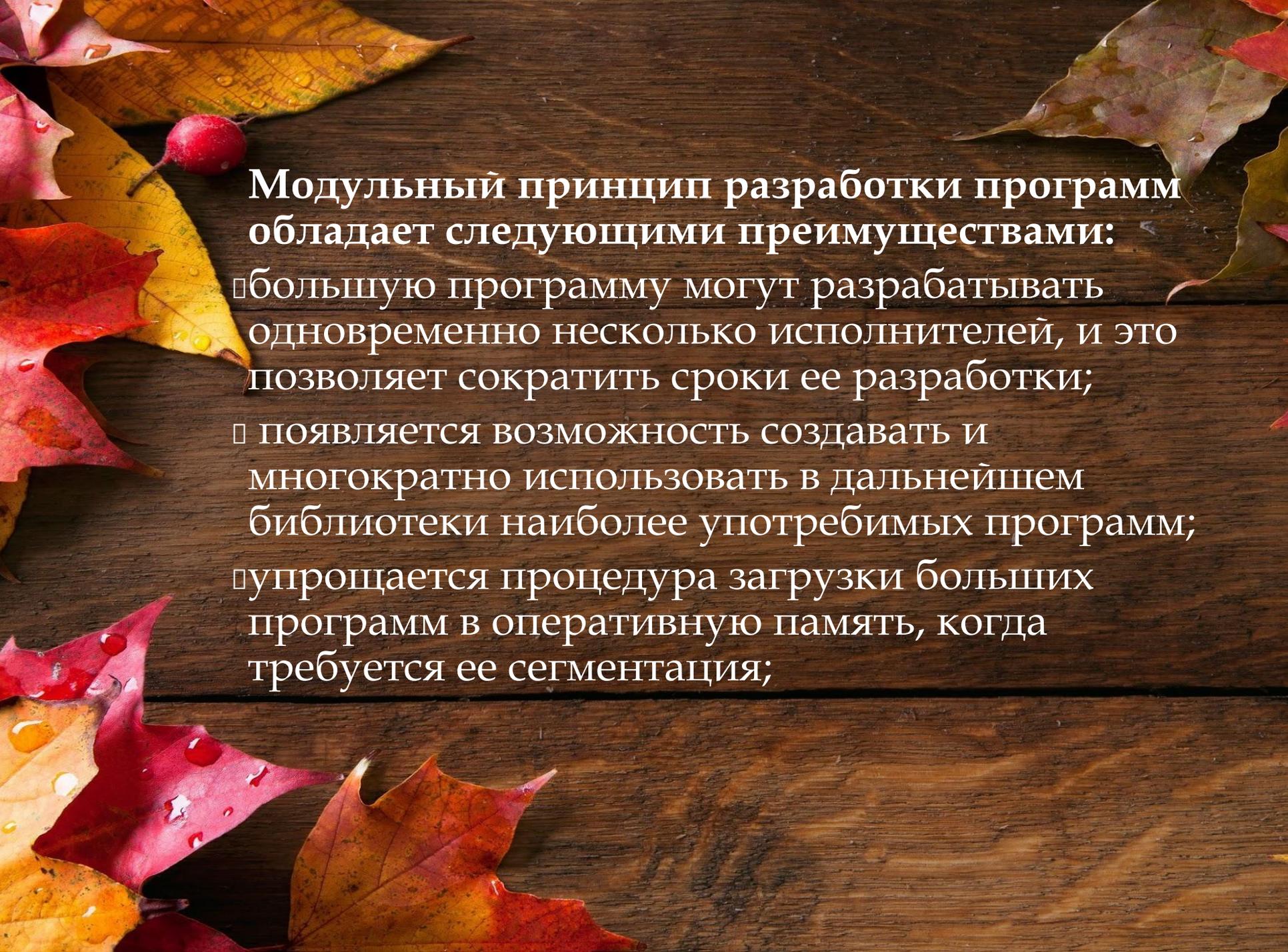
Программные модули должны обладать следующими свойствами:

- ▣ на модуль можно ссылаться (т.е. обращаться к нему) по имени, в том числе и из других модулей;
- ▣ по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;
- ▣ модуль должен иметь один вход и выход;
- ▣ модуль должен иметь небольшой размер, обеспечивающий его обзоримость.



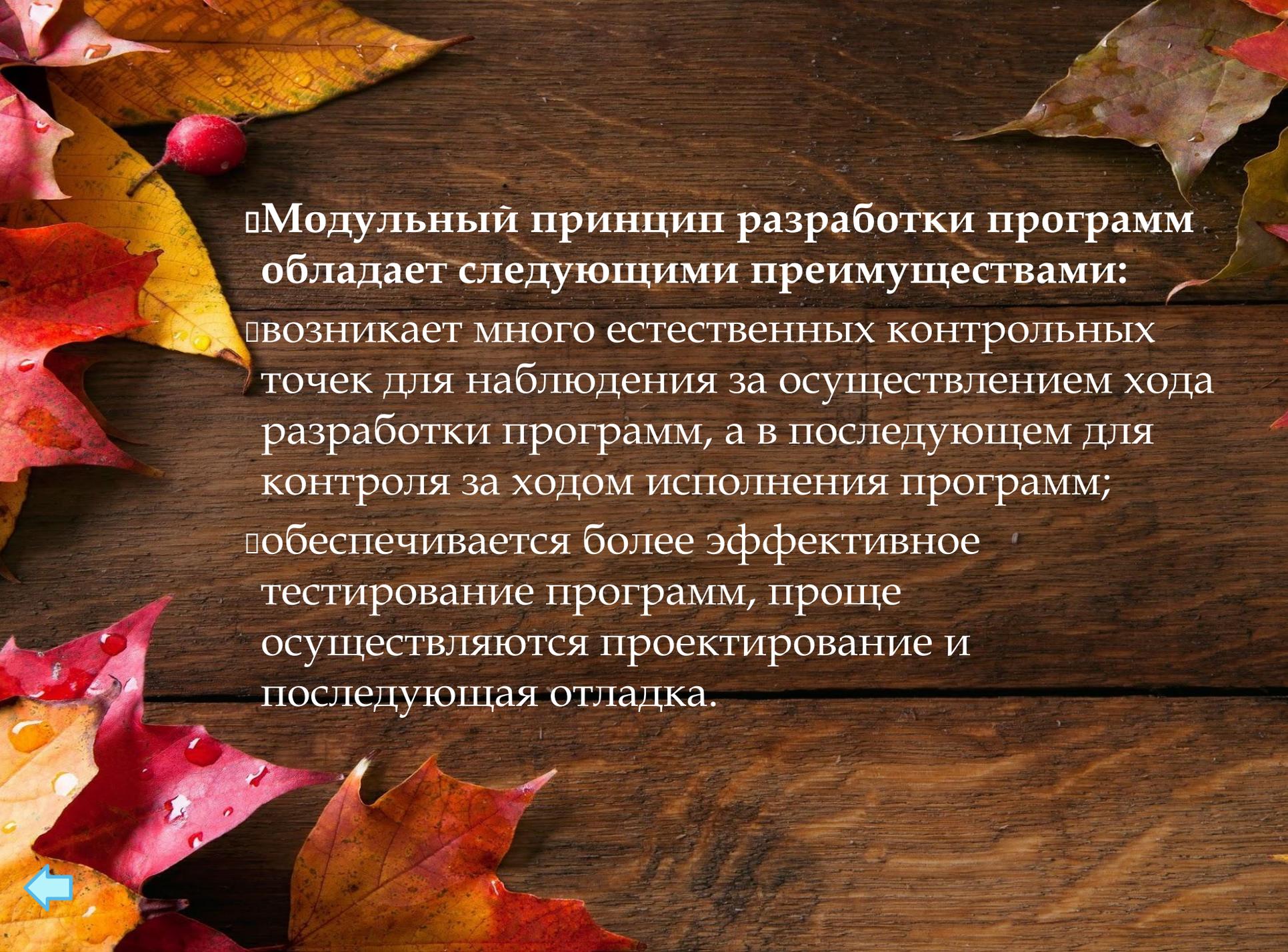
При разработке сложных программ в них выделяют головной управляющий модуль, подчиненные ему модули, обеспечивающие реализацию отдельных функций управления, функциональную обработку (т.е. непосредственную реализацию основного назначения программного комплекса), а также вспомогательные модули, обеспечивающие сервисное обслуживание пакета (например, сбор и анализ статистики работы программы, обработка различного рода ошибочных ситуаций, обучение и выдача подсказок и т.п.).





Модульный принцип разработки программ обладает следующими преимуществами:

- ▣ большую программу могут разрабатывать одновременно несколько исполнителей, и это позволяет сократить сроки ее разработки;
- ▣ появляется возможность создавать и многократно использовать в дальнейшем библиотеки наиболее употребимых программ;
- ▣ упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;



▣ Модульный принцип разработки программ обладает следующими преимуществами:

▣ Возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем для контроля за ходом исполнения программ;

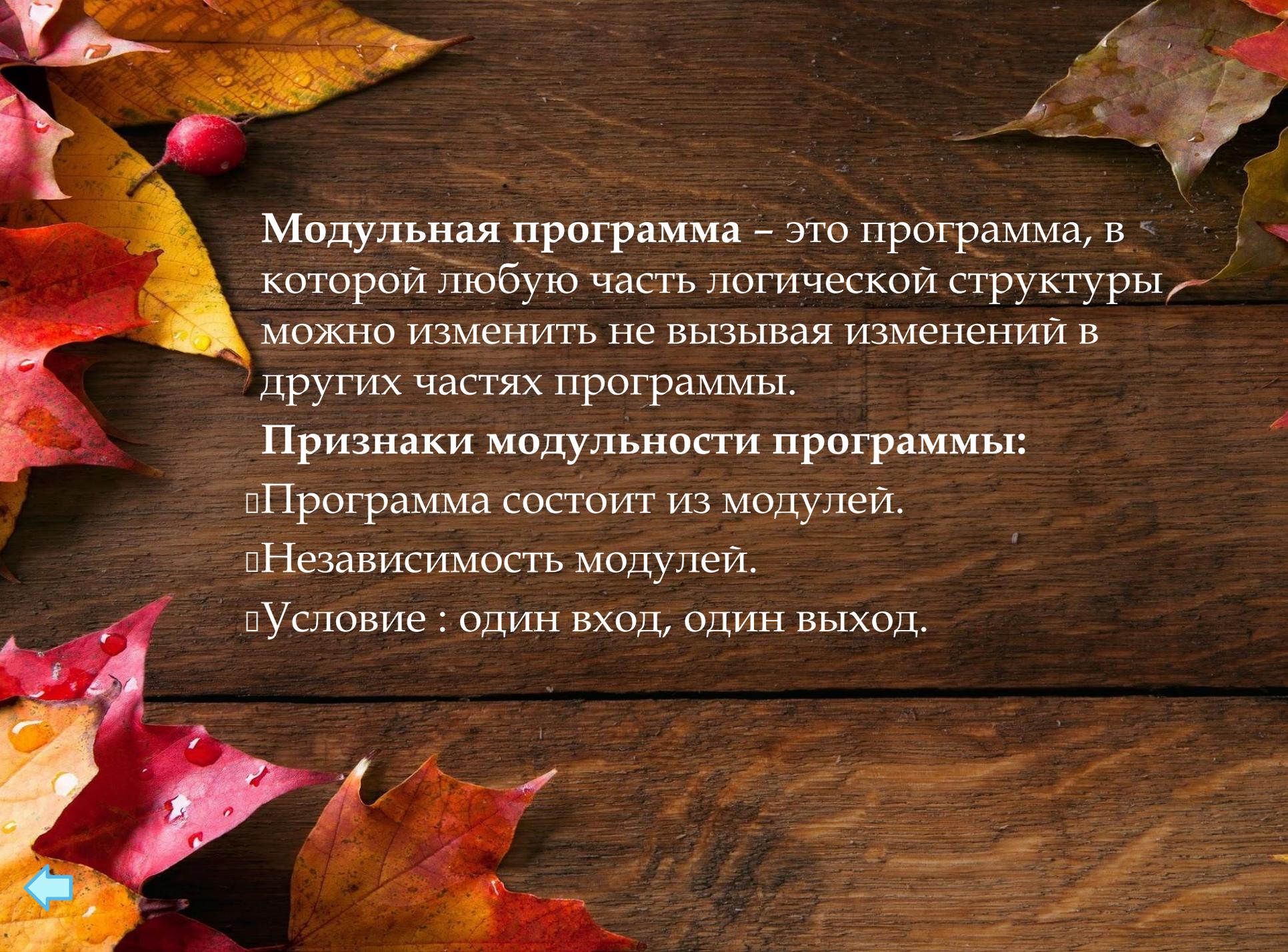
▣ Обеспечивается более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.



Недостатки:

- ▣модульность требует большей дополнительной работы программиста и определенных навыков проектирования программ.
- ▣модульный подход в среднем требует большего времени работы ЦП(на 5-10%) за счет времени обращения к модулям.
- ▣модульность программ приводит к увеличению объема программ(на 5-10%).

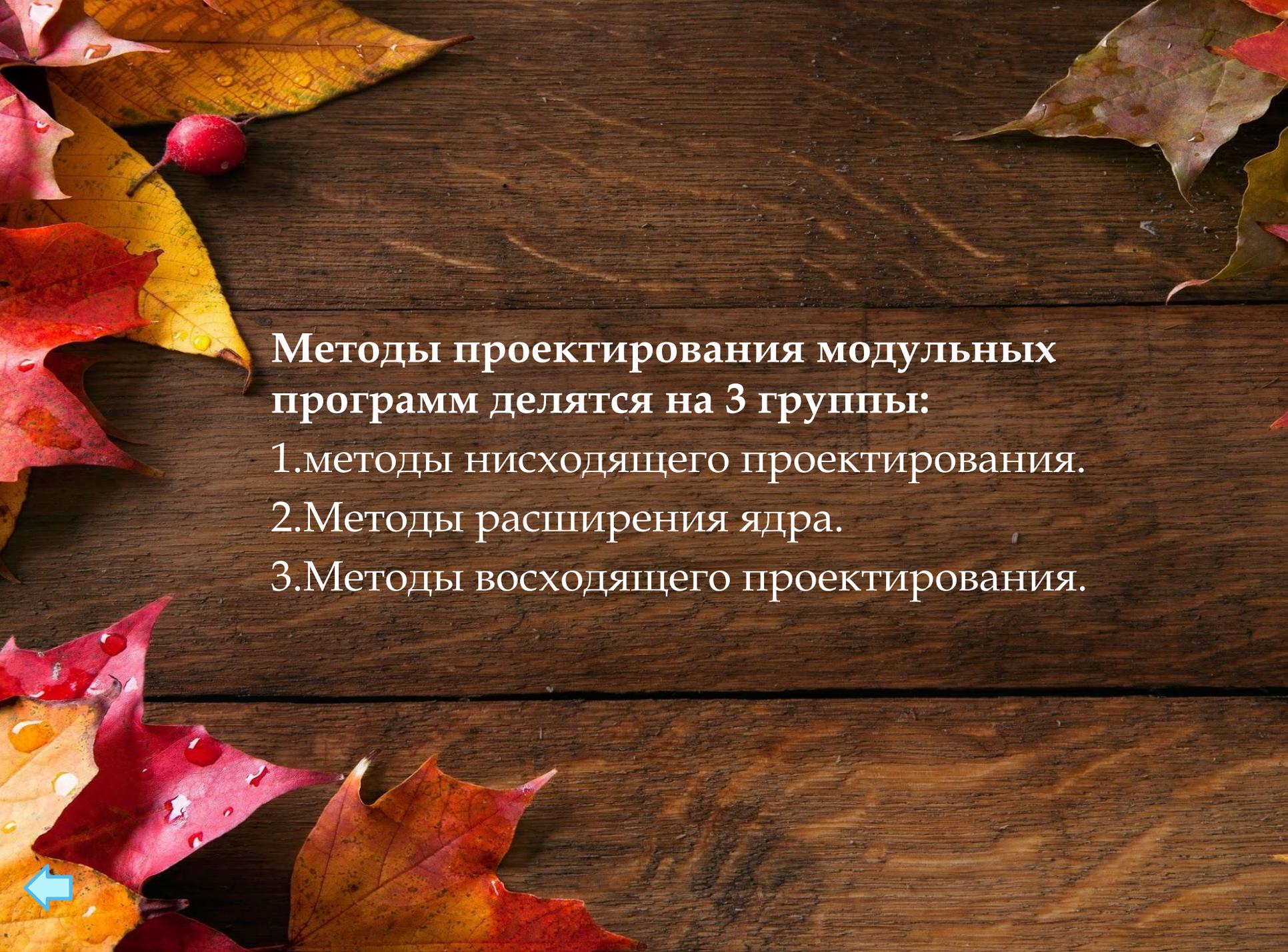




Модульная программа – это программа, в которой любую часть логической структуры можно изменить не вызывая изменений в других частях программы.

Признаки модульности программы:

- ▣ Программа состоит из модулей.
 - ▣ Независимость модулей.
 - ▣ Условие : один вход, один выход.
- 



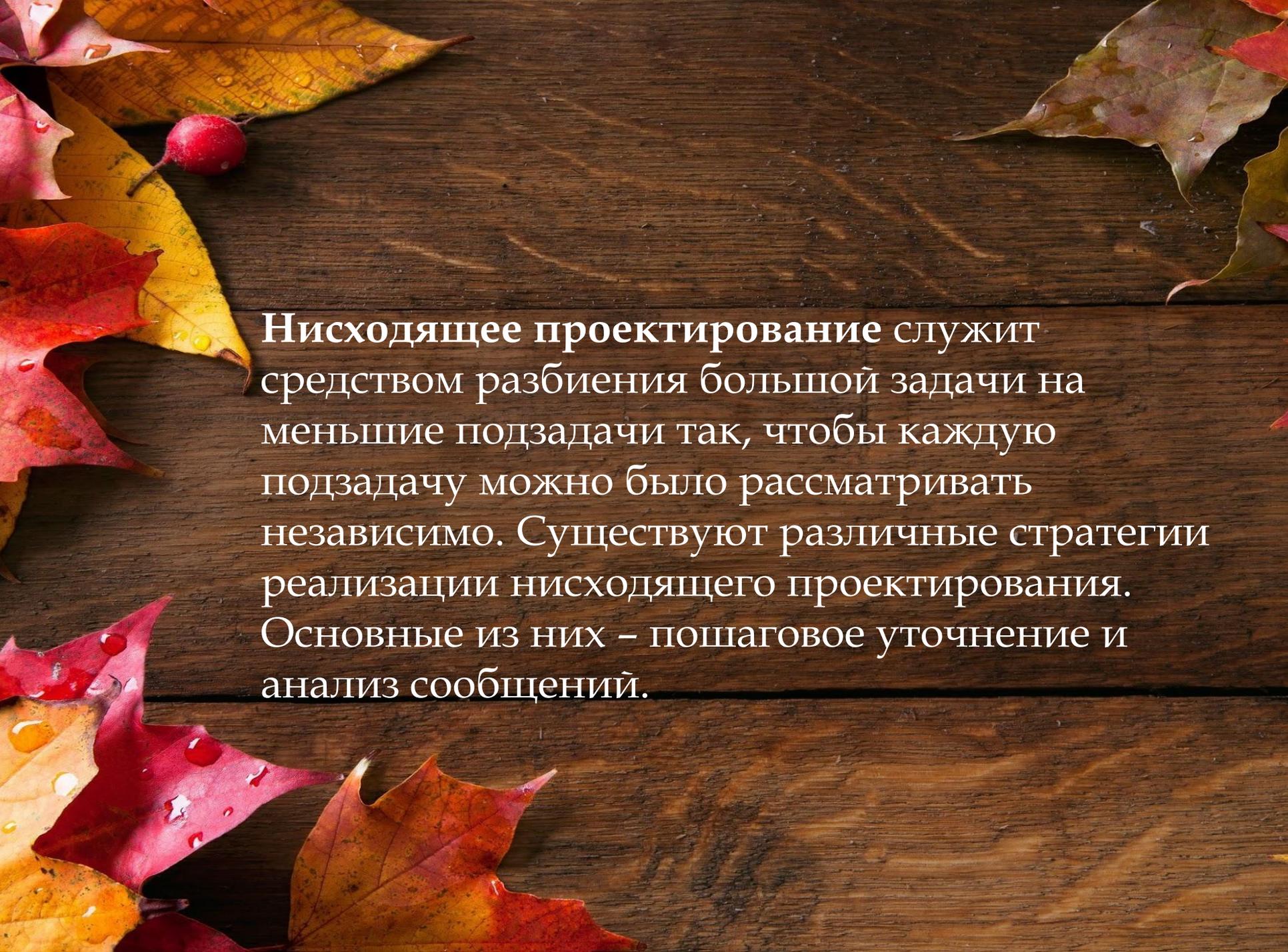
Методы проектирования модульных программ делятся на 3 группы:

1.методы нисходящего проектирования.

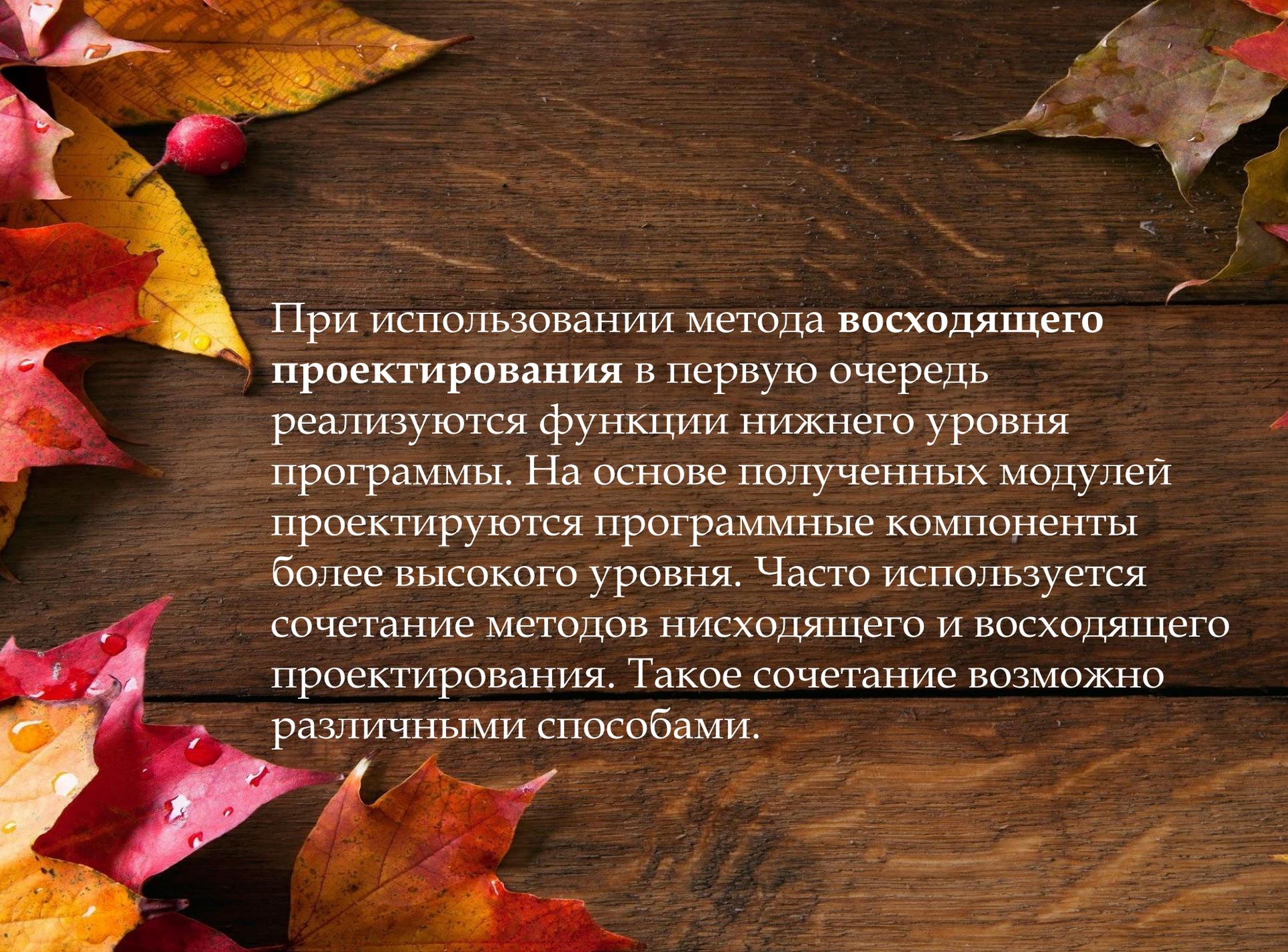
2.Методы расширения ядра.

3.Методы восходящего проектирования.

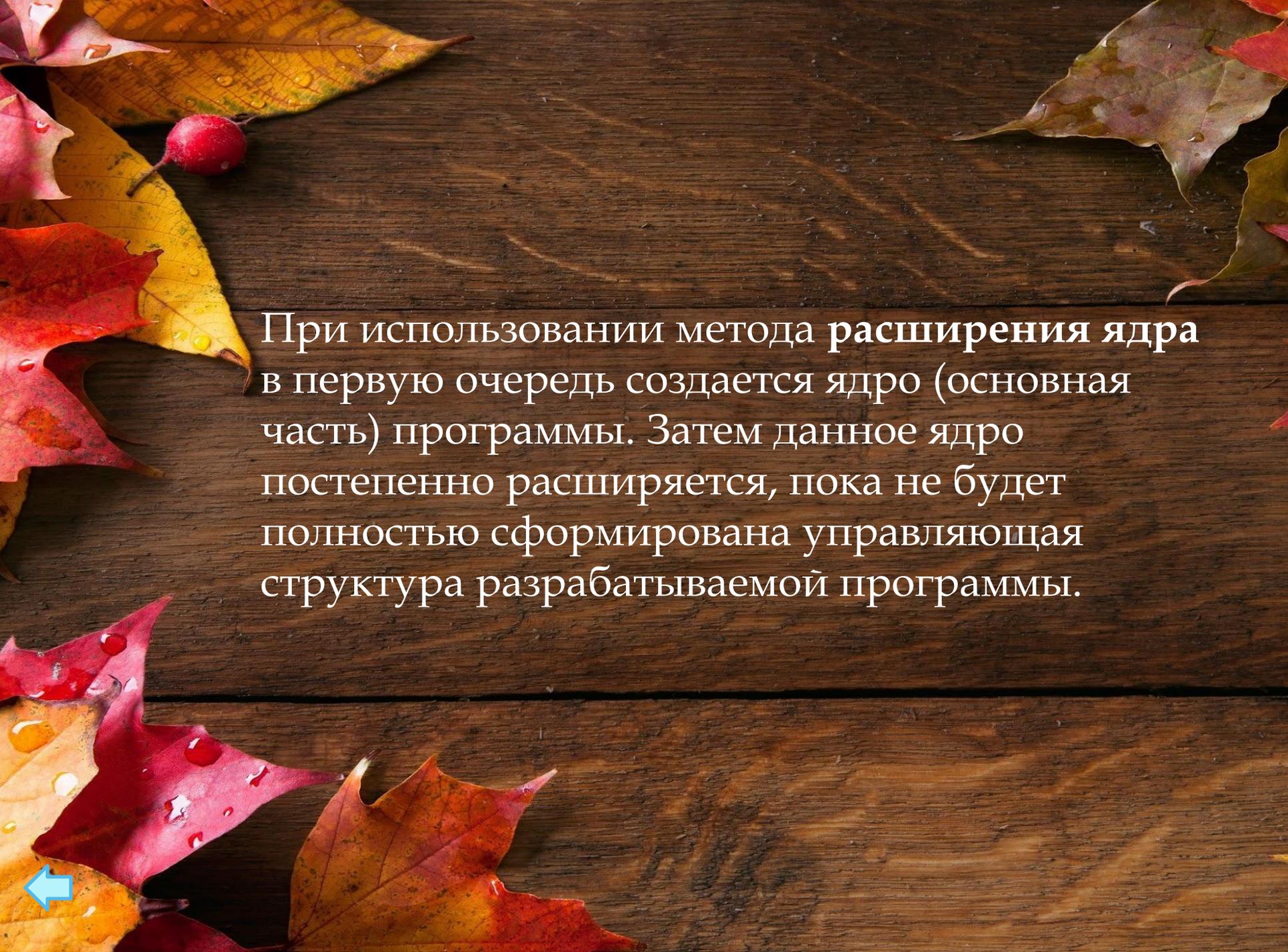


The background of the slide is a dark, textured wooden surface. Scattered around the edges are several autumn leaves in various colors: yellow, orange, red, and pink. Some leaves have small water droplets on them. A single red berry is visible in the upper left quadrant.

Нисходящее проектирование служит средством разбиения большой задачи на меньшие подзадачи так, чтобы каждую подзадачу можно было рассматривать независимо. Существуют различные стратегии реализации нисходящего проектирования. Основные из них – пошаговое уточнение и анализ сообщений.

The background of the slide is a dark, textured wooden surface. Scattered around the edges are several autumn leaves in various colors: yellow, orange, red, and pink. Some leaves have small, clear water droplets on them. In the upper left corner, there is a small, round, red berry on a short stem.

При использовании метода **восходящего проектирования** в первую очередь реализуются функции нижнего уровня программы. На основе полученных модулей проектируются программные компоненты более высокого уровня. Часто используется сочетание методов нисходящего и восходящего проектирования. Такое сочетание возможно различными способами.



При использовании метода **расширения ядра** в первую очередь создается ядро (основная часть) программы. Затем данное ядро постепенно расширяется, пока не будет полностью сформирована управляющая структура разрабатываемой программы.

