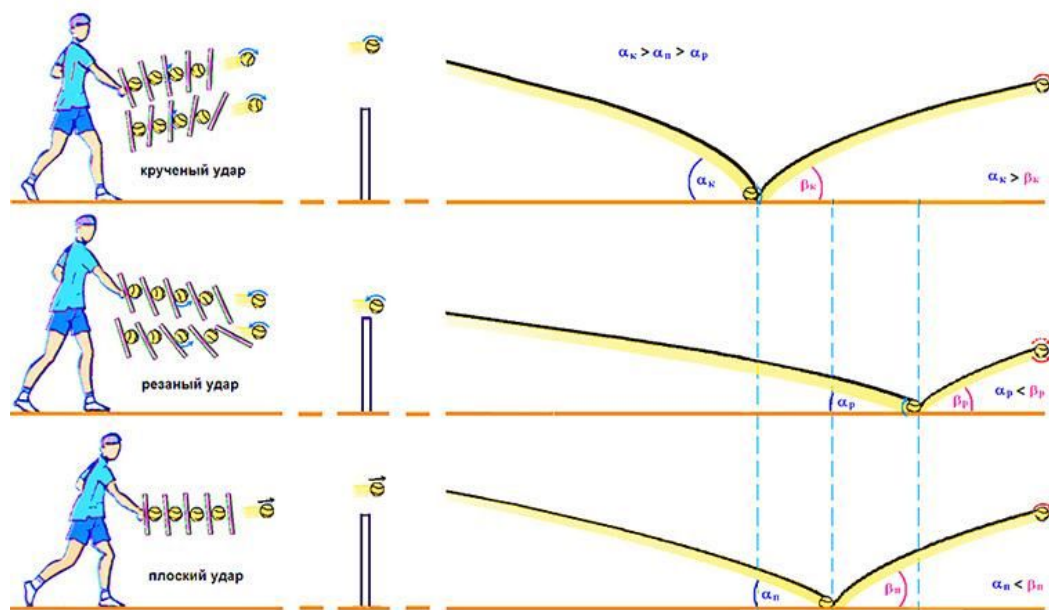

2.15. Системы и технологии программирования

1. Этапы реализации задачи на ЭВМ:

- 1. Содержательная постановка задачи
- 2. Математическая постановка задачи
- 3. Формализация математической постановки
- 4. Разработка алгоритма решения задачи
- 5. Разработка ПО, реализующего алгоритм
- 6. Отладка ПО
- 7. Тестирование ПО и анализ результатов
- 8. Проведение доработок по итогам тестирования
- 9. Разработка технической документации для пользователей
- 10. Сдача в эксплуатацию.

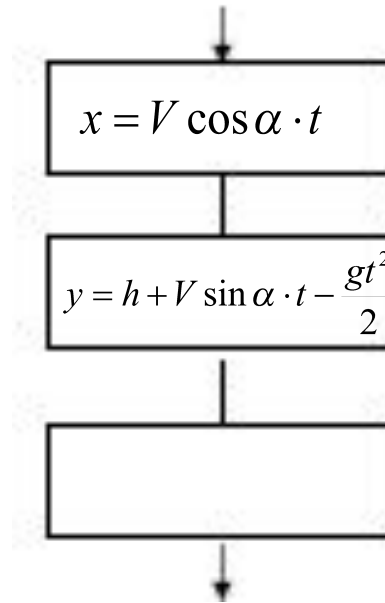
Формализация задачи состоит в том, чтобы построить математическую модель, т. е перейти от содержательной постановки задачи к ее строгому формальному описанию. В процессе формализации реальным объектам присваиваются математические наименования, а связи между ними описываются математическими зависимостями.



$$x = V \cos \alpha \cdot t,$$

$$y = h + V \sin \alpha \cdot t - \frac{gt^2}{2}$$

Разработка алгоритма решения задачи предполагает ~~с помощью математической модели определить и~~ описать совокупность и последовательность действий, ведущих к нахождению требуемого результата.



Программирование на алгоритмическом языке необходимо для ввода алгоритма в память ПЭВМ.

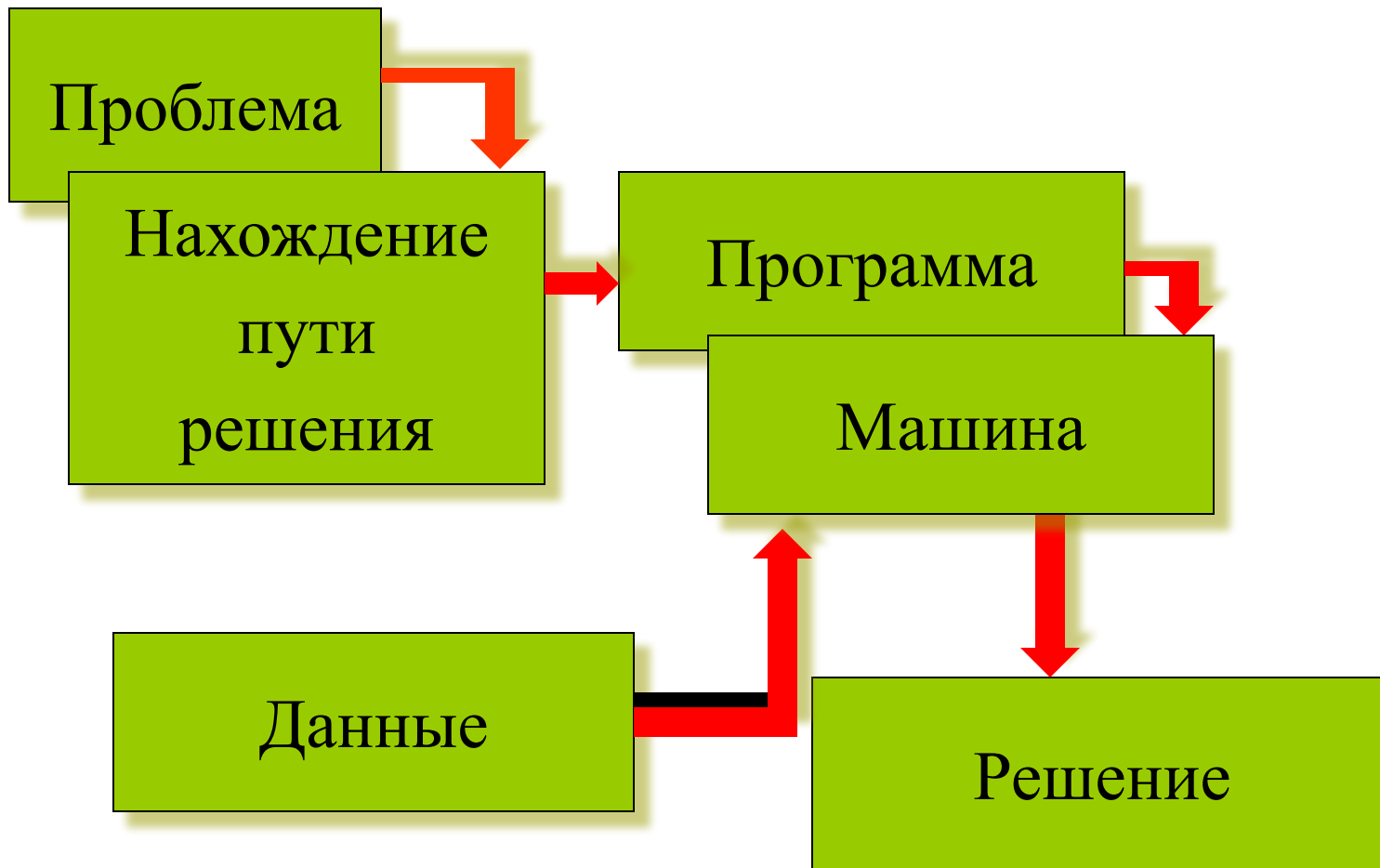
```
a := 100;  
b := 30;  
a := a - b * 3;  
if a > b then  
  c := a - b  
else c := b - a;
```

Трансляция программы - преобразование программы с алгоритмического языка на язык машины.

```
8A 16 4A 00  10001010 00010110 01001010 00000000  
FE CA      11111110 11001010
```

Отладка программы заключается в проверке ее работоспособности. В процессе отладки задача с контрольными данными решается на ПЭВМ.

Для того, чтобы компьютер смог получить нужный результат, ему необходимо указать путь решения, то есть задать программу и, возможно, исходные данные.



2. Система программирования ПЭВМ.

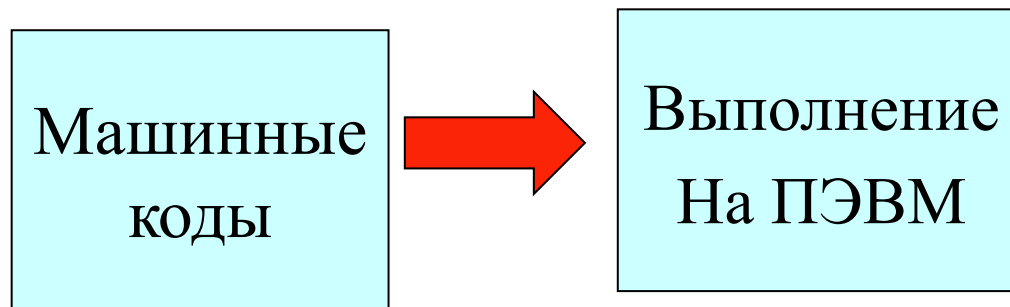
Три компонента системы программирования:

1. язык программирования,
2. транслятор (интерпретатор или компилятор)
3. библиотека программ.

2. Система программирования ПЭВМ.

– На начальном этапе развития вычислительной техники программирование осуществлялось только в машинных командах.

При программировании в машинных командах программист находится как бы на одном уровне с ПЭВМ: между его программой и реализующими ее техническими средствами ПЭВМ нет никаких посредников и программа выполняется практически в том виде, в каком ее записал программист.



Противоречие между огромными возможностями ПЭВМ, способных выполнять сотни тысяч команд в секунду, и столь же огромными временными затратами на разработку и отладку программ.

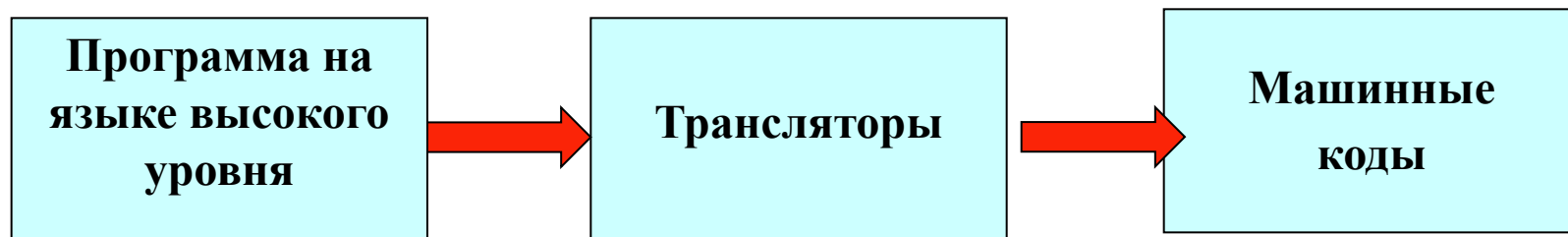


Языки
Ассемблера

Основное отличие языка ассемблера от машинного языка заключается в **использовании вместо двоичных кодов операций и адресов их символических обозначений.**

Далее появились языки высокого уровня. Команды языка высокого уровня - слова естественного языка, что упрощает работу программиста.

Чтобы вычислительная машина могла выполнить программу, написанную на каком-либо языке программирования, в её программном обеспечении **должна быть программа - транслятор для этого языка.**



Транслятор превращает программу в исполняемый файл

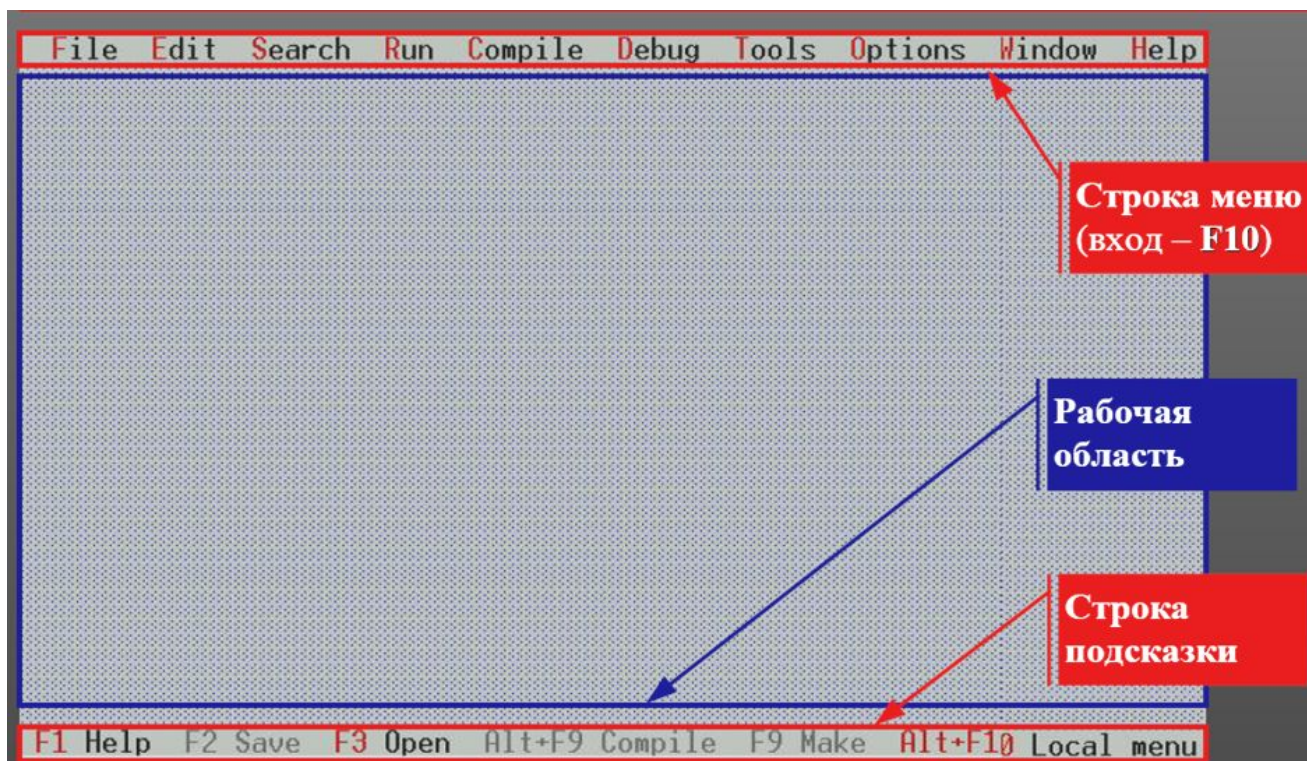
Трансляторы бывают двух типов:

- ❖ *интерпретаторы*
- ❖ *компиляторы*

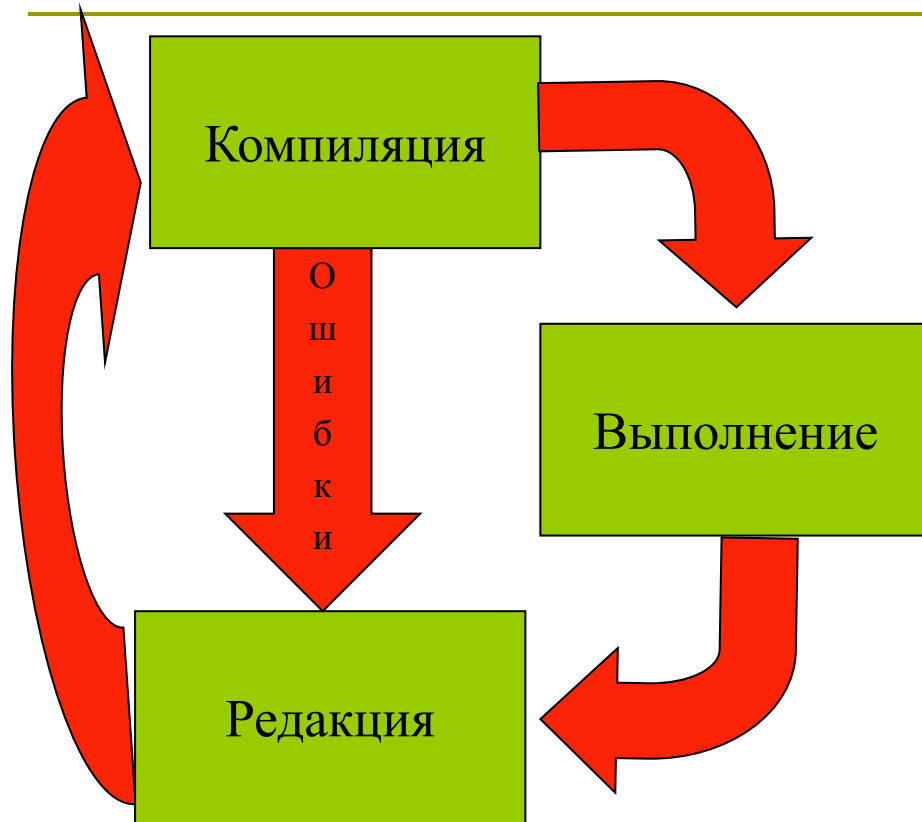
Интерпретатор - переводит каждую команду программы с одновременным её выполнением и, если обнаруживает ошибку, сообщает о ней и прекращает выполнение программы.

Компилятор - переводит всю программу целиком и в конце работы выдаёт список ошибок, если они обнаружены.

Система программирования это система автоматизации программирования, образуемая языками программирования, компиляторами и интерпретаторами программ, представленными на этих языках, соответствующей документацией, а также вспомогательными средствами для подготовки программ к выполнению.



Компилятор переводит исходный текст с языка высокого уровня на машинный язык, доступный компьютеру.



При выполнении программы могут быть ошибки. Для их поиска используются специальные средства, помогающие выявить ошибки - программа, называемая отладчиком.

Для ввода текстов в компьютер и их изменения используется программа, называемая редактором.

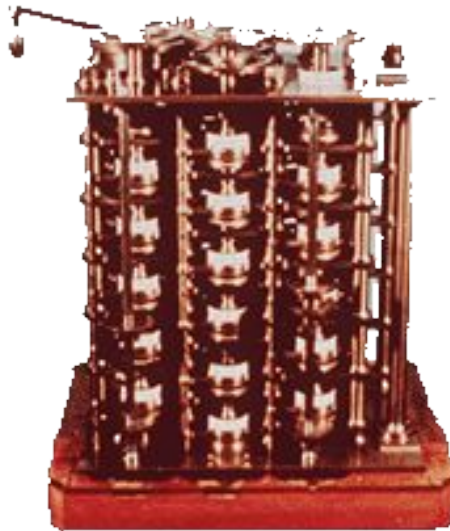
Системы программирования:

- 1. машинно-ориентированные
- 2. алгоритмические
- 3. проблемно-ориентированные
- 4. универсальные
- 5. ЯП для создания ПО

3. Назначение языков программирования

— Программирование появилось задолго до появления не только первых персональных ЭВМ, но вообще ЭВМ. (Программы леди **Ады Августы Лавлейс** для аналитической вычислительной машины **Бэббиджа**).

Языком программирования ЭВМ первого поколения был **язык машинных кодов** (язык низкого уровня). Во втором поколении появились языки **высокого уровня**. Команды **языка высокого уровня** - слова естественного языка, что упрощает работу программиста.



— Запись текста программы на языке программирования называют кодированием, а отладка программы - тестированием.

Пример программы.

```
Program second;
```

```
Var num: integer; {раздел описания данных}
```

```
begin
```

```
  num:=1;
```

```
  write('Хорошее число', num)
```

```
{тело прог-}
```

```
{раммы }
```

```
end.
```


Языки программирования

Язык C - это самый распространённый язык программирования. На нём написано больше программ, чем на любом другом. Подавляющее большинство профессиональных программистов владеют им. Исторически этот язык неотделим от операционной системы UNIX, которая в наши дни переживает своё второе рождение. 60-е годы были эпохой становления операционных систем и языков программирования высокого уровня. Язык C с самого начала создавался так, чтобы на нём можно было писать системные задачи. Разработчики языка - Кеннет Томсон и Деннис Ричи.

Языки программирования

Основные концепции языка Pascal были разработаны в 1967 г. профессором Николаусом Виртом, и он быстро превратился из средства для обучения студентов программированию в инструмент для создания новых программных проектов. Язык назван в честь французского учёного Блеза Паскаля, внесшего вклад в развитие средств вычислительной техники.

Языки программирования

Delphi - это ни что иное, как **Visual Pascal**. Эта среда – программирования простая и удобная и **содержит большую библиотеку готовых компонентов**: кнопок, панелей, списков, таймеров, изображений и т.д.

Basic - это один из самых старых языков программирования. Его создатели - Джон Кемени и Том Куртц, работавшие в Дортмундском колледже в 1964 году. Свой язык они назвали по первым буквам слов «**Beginner's All Purpose Symbolic Instructions Code**». Интерпретатор **Basic** был первым программным продуктом фирмы **Microsoft**, основанной Полом Аленом и Уильямом Гейтсом в 1975 году. В дальнейшем он не только поставлялся как программа, но и зашивался в ПЗУ компьютеров.

В середине 80 - годов фирма **Microsoft** разработала **QuickBasic**. Это был уже компилятор, а не интерпретатор. Вообще **Basic'ов** несколько сотен. После появления **Windows** и визуальных средств разработки программ был создан **Visual Basic**.

Языки программирования

FORTRAN - старейший язык программирования. В начале 50-х годов он был разработан исследовательской группой под руководством Джона Бэкуса. Его название происходит от 2-х слов: **FORMULA TRANSLATION**. Первая версия системы **FORTRAN** для компьютера **IBM** была выпущена в начале 1957 г.

ALGOL - это также старейший язык программирования, универсальный язык программирования общего назначения. Его название происходит от 2-х слов: **ALGORITHMIC LANGUAGE**. Он был разработан международной рабочей группой, состоящей из 30 человек в 1969 г. Язык очень стройный и изящный благодаря тому, что он основан на принципе ортогональности, согласно которому язык должен по возможности состоять из независимых элементов, свободно комбинирующихся друг с другом. Один из активных разработчиков **ALGOL-68** - Чарльз Линдси.

Языки программирования

ADA - язык программирования, который по сложности сравнивают с C++. Назван в честь леди Ады Августы Лавлейс, работавшей вместе с Чарльзом Беббиджем и разрабатывавшей программы для его «аналитической машины». Разработан язык группой под руководством в 1979 г. Жана Ишбиа в рамках конкурса, объявленного Министерством Обороны США. Мода на объектно-ориентированное программирование привела к созданию новой принципиально его версии ADA95.

4. Технологии программирования

При написании программ пользуются некоторым набором методов, способов, принципов, которые в совокупности можно определить термином "технологические средства".

Принцип модульности

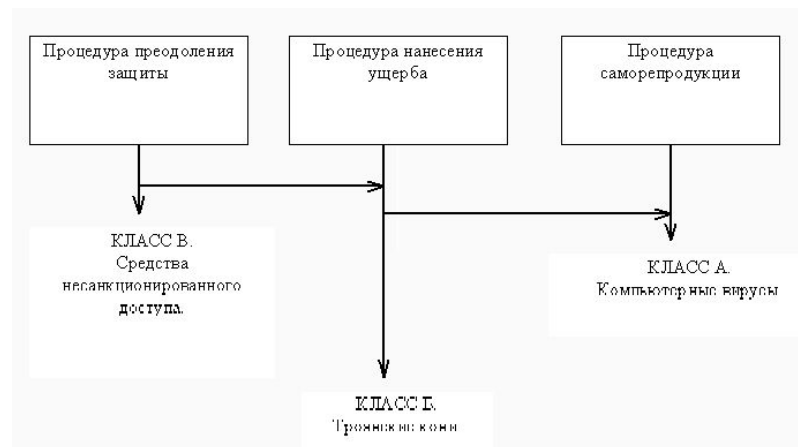
Стратегии проектирования

Метод структурного программирования

Принцип модульности

Модульность - это принцип программирования, состоящий в том, что большие и сложные программы разрабатываются и отлаживаются по частям, которые затем объединяются в единый комплекс.

Программа считается модульной только в том случае, если каждый входящий в нее модуль не зависит от других, т.е. его можно модифицировать, не вызывая каких либо изменений в других модулях.



Стратегии проектирования

Стратегия проектирования определяет "направление движения" программиста от исходной постановки задачи к конечной цели - программе решения этой задачи.

Различают две стратегии: нисходящее проектирование (сверху вниз) и восходящее (снизу вверх). В основе обеих стратегий лежит принцип модульности.

Стратегии проектирования

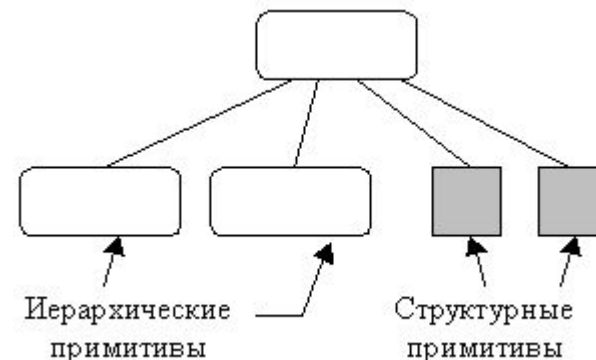
Нисходящее проектирование состоит в том, что разработка программы начинается сверху, с самых общих сведений о проекте и продолжается по уровням иерархии вниз путем разбиения на каждом уровне задачи предыдущего уровня на ряд более мелких, самостоятельных подзадач до тех пор, пока не окажется возможным записать программы решения задач нижнего уровня на одном из языков программирования.



Стратегии проектирования

Восходящее проектирование состоит в том, что за основу проекта берутся какие-то уже готовые программные модули, из которых строятся другие, более сложные или недостающие в исходном наборе, таким образом, чтобы в итоге выполнялась исходная постановка задачи.

Недостатком данной стратегии является то, что логически стройная структура программы получается лишь в результате многочисленных проб и ошибок, что затягивает сроки разработки программ.



Метод структурного программирования

Принцип метода структурного программирования состоит в том, что программа разрабатывается в виде определенным образом соединенных структур трех различных типов:

1. последовательная (линейная), в которой команды или программы выполняются последовательно в том порядке, в котором они записаны;
2. условная (выбора, или ветвления), в которой ход вычислительного процесса зависит от выполнения некоторого условия;
3. циклическая (повторения), в которой определенная последовательность команд или программ повторяется до тех пор, пока не перестанет выполняться некоторое условие.

Домашнее задание

- Стр. 105-110 учебник.
- Подготовиться к контрольному опросу