Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Описание самостоятельной работы студентов по дисциплине «Технологические подходы к разработке программного обеспечения»

Новиков Ф.А.,

канд. физ.-мат. наук, доцент кафедры «Технологии программирования»

Санкт-Петербург
2007

Оглавление

Введение

Самостоятельная работа студентов по курсу «Технологические подходы к разработке программного обеспечения» проводится в двух направлениях.

Первое направление — участие в панельных дискуссиях на учебном семинаре. Порядок подготовки и проведения семинара и требования к нему изложены в описании практической работы студентов по курсу «Технологические подходы к разработке программного обеспечения».

Второе направление самостоятельной работы студентов имеет другую цель – углубить и закрепить знания и навыки студентов именно в объеме материала, охваченного лекционным курсом.

С этой целью в данном пособии по каждой теме, входящей в курс, дается небольшое резюме, приводится оглавление темы и вопросы для самопроверки. Резюме призвано напомнить студенту об основных идеях и особенностях темы, оглавление может помочь сориентироваться в материале.

Вопросы для самопроверки принадлежат к одному из трех видов:

- вопросы по определениям;
- вопросы по фактам;
- вопросы «ловушки».

Вопрос по определению предполагает в качестве ответа точное и развернутое определение некоторого понятия. Такое определение, как правило, в явном виде приведено в учебно-методическом пособии по курсу.

Вопрос по факту предполагает указание конкретного факта. Как правило, в тексте учебно-методического пособия приведен либо ответ, либо ссылка на литературу, в которой можно найти ответ на вопрос по факту.

Вопрос «ловушка» требует некоторого размышления, ответ на такой вопрос требует понимания материала и творческое применение определений, а не только их механического запоминания. Основным инструментом поиска ответа на такой вопрос является здравый смысл учащихся.

Тема 1. Технология программирования

Совокупность известных технологических подходов к разработке программного обеспечения, или, несколько короче, технология программирования, является инженерной дисциплиной, входящей в обязательный набор знаний и умений всякого инженера, причастного к созданию и эксплуатации программного обеспечения компьютеров. Технология программирования имеет четко выделенный объект изучения – процессы разработки и сопровождения программного обеспечения, но, в настоящее время, не имеет единого метода и общепринятого способа построения. Технология программирования не является строгой математической дисциплиной, которую можно изложить последовательно, начиная с основополагающих понятий и применяя дедуктивные доказательства. Напротив, технология программирования является собранием разнородных и часто несогласованных друг с другом моделей, методик и средств. Кроме того, технологические приемы разработки программного обеспечения очень быстро меняются, почти каждые полгода предлагаются новые подходы.

В первой теме рассматривается сам предмет дисциплины – технология программирования – с самых общих позиций. Здесь же вводится разделение технологии программирования на три составляющих: модель процесса, модель команды и дисциплину программирования.

В первой теме рассматриваются следующие вопросы.

1.1. Назначение технологии программирования

1.2. История развития технологии программирования

1.2.1. Дореволюционный период

1.2.2. «Революция в программировании»

1.2.3. Послереволюционный период

1.3. Типы программных проектов

1.4. Составные части технологии программирования

1.5. Проект, продукт, процесс и персонал

Вопросы для самопроверки

1. Что такое программирование?
2. Можно ли сказать, что процесс набора номера на мобильном телефоне для вызова абонента является программированием?
3. Что вы понимаете под термином «технология программирования»?
4. Является ли процесс набора номера на мобильном телефоне для вызова абонента предметом технологии программирования?
5. Что вы понимаете под термином «информатика»?
6. Что общего и в чем различие между информатикой и технологией программирования?
7. На какие периоды можно разделить историю развития технологии программирования?
8. Для каких целей применялись компьютеры в первый период своего существования?
9. Какие основные технологические идеи появились в первый период развития программирования?
10. Какие изменения произошли в сфере применения компьютеров в середине 60-х годов?
11. Какие проблемы привели к кризису программирования в этот период?
12. Перечислите фундаментальные идеи технологии программирования, сформировавшиеся в это время.
13. Какое развитие эти идеи получили в дальнейшем?
14. По каким факторам можно классифицировать программы?
15. Какую технологию программирования можно назвать наилучшей?
16. Перечислите различные аспекты технологии программирования.
17. Что такое «модель процесса»?
18. Что такое «модель команды»?
19. Что такое «дисциплина программирования»?

20. Перечислите четыре «П» технологии программирования. Как они связаны между собой?

Рекомендуемая литература по теме

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.

2. Брукс-мл. Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1975; новое издание перевода: Мифический человеко-месяц. СПб.: СИМВОЛ+, 1999.

3. Орлов С. Технологии разработки программного обеспечения. – СПб.: Питер, 2002.

3. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

Тема 2. Жизненный цикл программы

Понятие жизненного цикла программы является ключевой абстракцией, вокруг которой строятся все известные в настоящее время технологические подходы к программированию.

Жизненный цикл программы – это некоторая абстрактная модель, однако элементы принимаемого жизненного цикла являются тем материалом, из которого стоятся различные конкретные модели технологии программирования. Поэтому обсуждение понятия жизненного цикла программы необходимо должно предшествовать обсуждению элементов всякой технологии программирования.

Предложено несколько альтернативных моделей жизненного цикла. Они близки в своей принципиальной основе, но различаются, причем существенно, в деталях. Эти различия являются тем питательным материалом, на котором вырастают все новые технологические подходы к разработке программного обеспечения.

Во второй теме рассматриваются следующие вопросы.

- 2.1. Циклический характер разработки
- 2.2. Основные понятия технологии программирования
  - 2.2.1. Процессы и модели
  - 2.2.2. Фазы и витки
  - 2.2.3. Вехи и артефакты
  - 2.2.4. Заинтересованные лица и работники
- 2.3. Выявление и анализ требований
  - 2.3.1. Требования к программному обеспечению
  - 2.3.2. Схема разработки требований
  - 2.3.3. Управление требованиями
- 2.4. Архитектурное и детальное проектирование
  - 2.4.1. Архитектурное проектирование
  - 2.4.2. Детальное проектирование
- 2.5. Реализация и кодирование
- 2.6. Тестирование и верификация
  - 2.6.1. Процесс контроля качества

Вопросы для самопроверки

1. Что такое «жизненный цикл программы»?
2. Что такое «выпуск»?
3. Имеет ли жизненный цикл программы начало и конец?
4. Каковы характеристики типичной отечественной программирующей организации?
5. Каковы основные действующие лица в жизненном цикле программы?
6. Что является предметом технологии программирования?
7. Как вы понимаете понятие «процесс»?
8. Что такое «модель»?
9. Какие средства используются для описания процесса программирования, и каковы основные структурные составляющие такого описания?
10. Что такое «фаза», чем она характеризуется?
11. Является фаза состоянием процесса или действием?
12. Какие фазы могут быть включены в модель процесса разработки?
13. Что такое «виток»? Имеет ли он что-либо общее с итерацией?
14. Что такое «веха» и «артефакт»? Приведите примеры.
15. Что такое «инструментальная программа»?
16. Что общего и в чем состоит различие в употреблении терминов «работник» и «роль?»
17. Чем характеризуется работник в рамках конкретного проекта?
18. Какова первая фаза жизненного цикла разработки программного обеспечения?
19. Как можно определить понятие «требование»?
20. Является ли утверждение «у обычного человека десять пальцев на руках, поэтому клавиатура мобильного телефона содержит не менее десяти кнопок» требованием?
21. Является ли утверждение «для записи номера телефона используется десять арабских цифр, поэтому клавиатура мобильного телефона содержит не менее десяти кнопок» требованием?

22. Какие решения могут применяться при формулировке требований?

23. Перечислите основные работы, которые выполняются при разработке требований.

24. Возможна ли разработка программного обеспечения без разработки требований? Обоснуйте ответ.

25. Могут ли измениться требования к разрабатываемой программной системе?

26. Что можно отнести к действиям по управлению требованиями?

27. Какими атрибутами можно однозначно идентифицировать требование?

28. Какие состояния требования вы можете назвать?

29. На какие части можно разделить фазу проектирования?

30. Какие типы программных архитектур вы знаете?

31. Чем можно характеризовать фазу детального проектирования?

32. Что такое «реализация» и что такое «кодирование»?

33. Что такое «инженерный анализ программы»?

34. Для чего нужны стандарты кодирования?

35. Что такое тестирование и контроль качества, какие методы и средства контроля вы знаете?

36. Что является целью тестирования?

37. Что такое «верификация», «валидация» и «системное тестирование»?

38. Какие виды системного тестирования вы знаете?

39. Что общего и в чем различие между сопровождением программы и продолжающейся разработкой?

Рекомендуемая литература по теме

1. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. – Спб.: Питер, 2004.

2. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.

3. Брукс-мл. Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1975; новое издание перевода: Мифический человеко-месяц. СПб.: СИМВОЛ+, 1999.

4. Орлов С. Технологии разработки программного обеспечения. – СПб.: Питер, 2002.

5. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

6. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб : Питер, 2002.

Тема 3. Модели процесса разработки

Трудности разработки реальных приложений обусловлены их сложностью, и критическую роль в преодолении этой сложности играет сам процесс разработки.

Существует необозримое множество конкретных процессов разработки, практически в каждой программирующей организации, а иногда и в каждом проекте используется свой процесс разработки. Однако, как правило, все конкретные процессы следуют некоторым общим принципам и имеют много общего.

Абстрактное описание общих принципов, используемых в некотором классе сходных проектов, называется моделью процесса разработки.

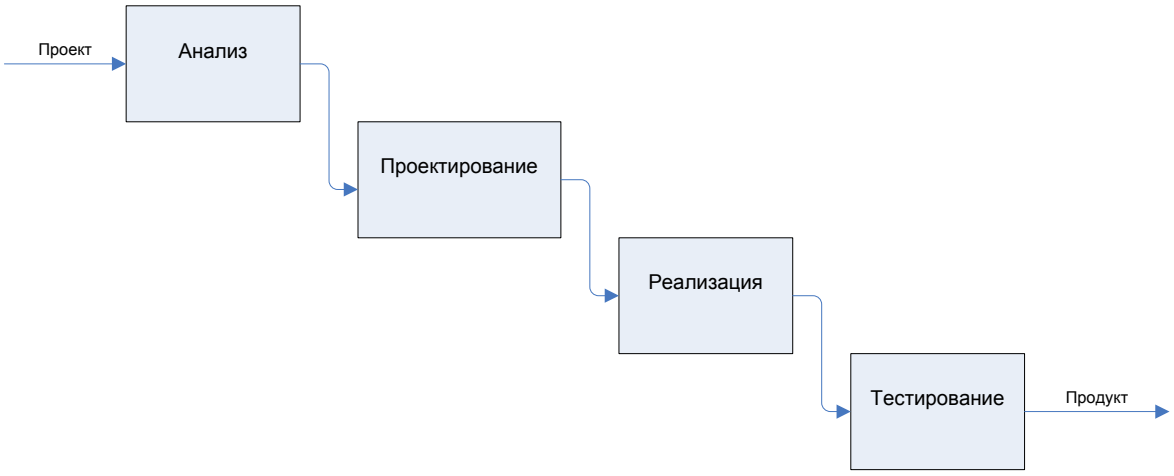
Моделей процесса разработки предложено довольно много (хотя и меньше, чем конкретных процессов, разумеется). Разнообразные модели также можно сгруппировать и классифицировать в соответствии с теми основными допущениями, которые принимаются при конструировании модели. Выделенные таким образом классы моделей принято называть стратегиями конструирования моделей процесса разработки.

В третьей теме рассматриваются следующие вопросы.

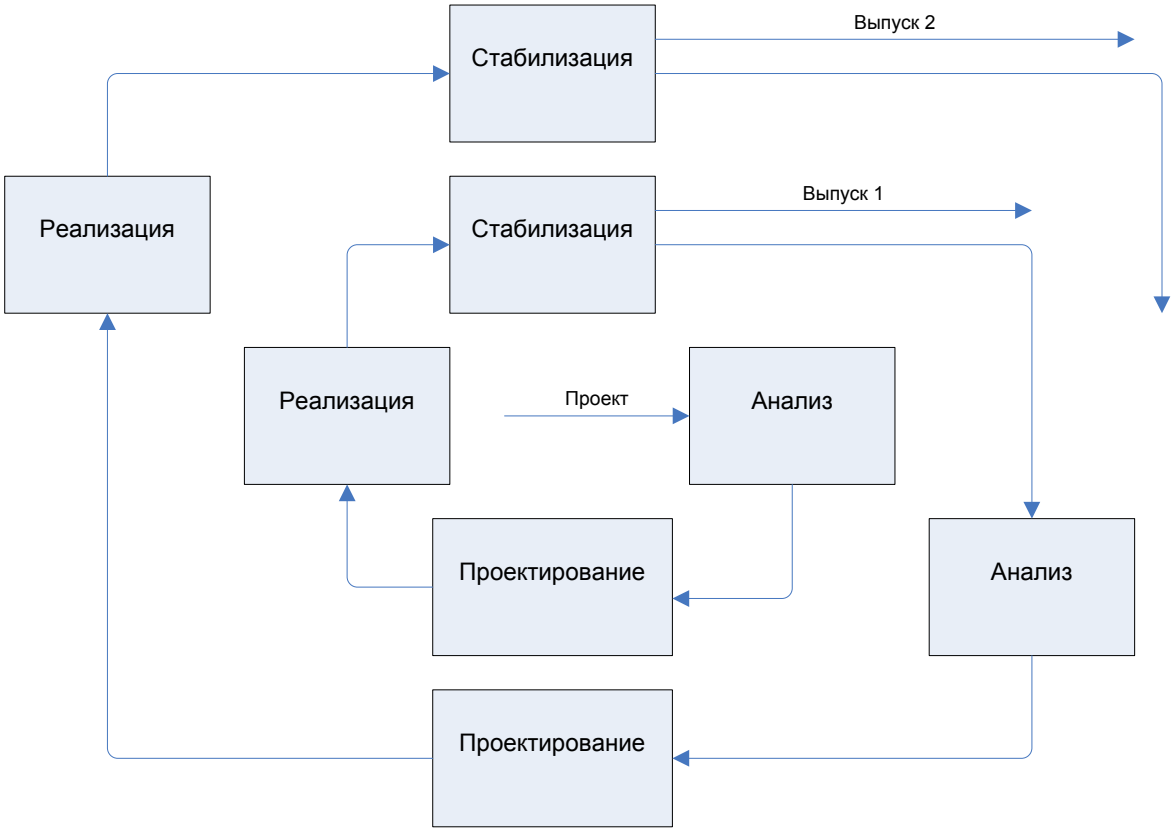
- 3.1. Водопадные и конвейерные модели
- 3.2. Спиральные и инкрементные модели
- 3.3. Гибкие модели процесса разработки
- 3.4. Конструирование модели процесса
  - 3.4.1. Выявление требований к процессу
  - 3.4.2. Используемые фазы, вехи и артефакты
  - 3.4.3. Выбор архитектуры процесса
  - 3.4.4. Порядок проведения типового проекта
  - 3.4.5. Документированные процедуры

Вопросы для самопроверки

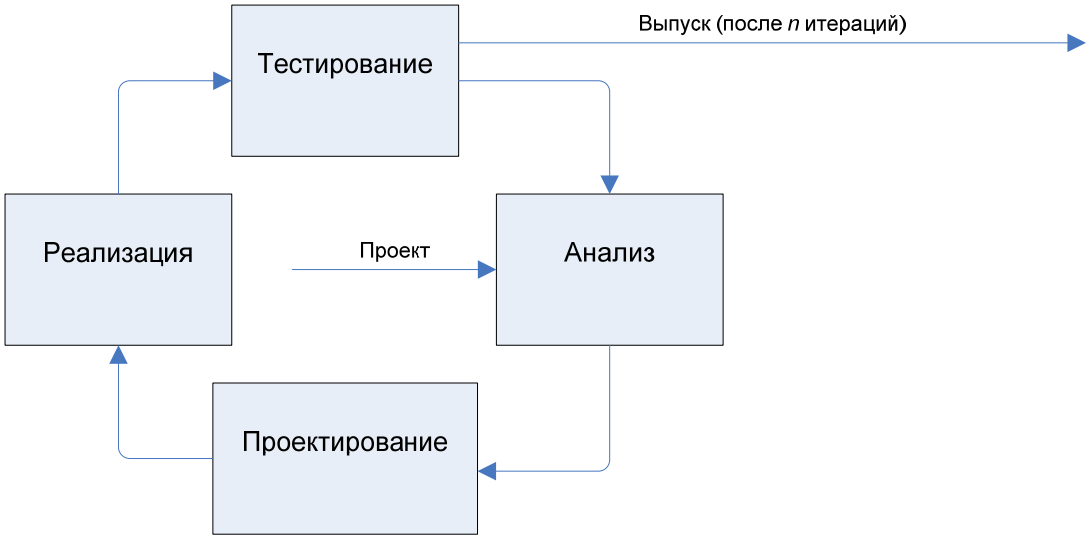
1. Какие стратегии конструирования модели процесса вы знаете?
2. Чем характеризуется линейная стратегия конструирования процесса разработки?
3. Чем характеризуется итерационная стратегия конструирования процесса разработки?
4. Чем характеризуется эволюционная стратегия конструирования процесса разработки?
5. Чем характеризуется модель водопада и модель конвейера?
6. Чем отличаются спиральные и инкрементные модели?
7. К какой стратегии разработки относится экстремальное программирование?
8. Является ли экстремальное программирование жестко фиксированным процессом разработки?
9. Перечислите ключевые методики экстремального программирования.
10. Какого эффекта можно ожидать от выборочного применения методик экстремального программирования? Приведите примеры.
11. Является ли сверхурочная работа положительным фактором? Обоснуйте ответ.
12. К какой модели процесса относится следующая схема?



13. К какой модели процесса относится следующая схема?



14. К какой модели процесса относится следующая схема?



15. Приведите пример требований к конструируемой модели процесса.

16. Включаются ли в число требований свойства, которыми конструируемый процесс не должен обладать?

17. Какие фазы используются в разрабатываемом процессе?

18. Всегда ли фазы одного витка всегда выполняются в указанном порядке?

19. Чем заканчивается каждая фаза?

20. Как называется главная веха фазы анализа?

21. Какие обязательные разделы должна включать в себя концепция проекта? На какой фазе концепция проекта должна появиться?

22. Какие еще термины применяются для обозначения концепции проекта?

23. Какие промежуточные вехи могут быть запланированы на фазе анализа?

24. Какая фаза является решающей для общего успеха проекта? Какова ее главная веха? Перечислите обязательные и важнейшие ее составляющие. Какие еще компоненты могут входить, кроме обязательных?

25. Что можно отнести к внутренним, а что следует отнести к внешним материалам?

26. Какова основная веха фазы реализации? Перечислите возможные промежуточные вехи фазы реализации.

27. Какова главная веха фазы стабилизации?

28. Какие виды комплексного тестирования вы знаете?

29. Каковы типичные приемы, применяемыми при тестировании устойчивости?

30. Какова цель тестирования функциональности?

31. Какие параметры проверяются при тестировании применимости?

32. Перечислите возможные промежуточные вехи фазы стабилизации.

33. Как называется заключительная фаза и какова ее главная веха?

34. Чем отличается фаза опытной эксплуатации от фазы внедрения?

35. Перечислите типичные вехи фазы внедрения.

36. Чем характеризуются легкий и сверхлегкий проекты? В чем их отличие от тяжелого?

37. Каковы могут быть критерии типизации проекта?

38. На чем основан критерий сложности?

39. Какова сравнительная продолжительность сверх легкого, легкого, тяжелого и сверх тяжелого проекта?

40. Какую структуру имеет модель процесса сверх легкого проекта?

41. Выполняются ли параллельно какие-либо фазы в модели процесса сверх легкого проекта? Чем это вызвано?

42. Приведите схематический план-график модели процесса легкого проекта.

43. Какую структуру имеет модель процесса тяжелого проекта?

44. Какие фазы в тяжелом проекте следует удлинить при планировании?

45. Приведите схематический план-график модели процесса тяжелого проекта.

46. Неудача на какой фазе тяжелого проекта переводит проект в категорию сверх тяжелых?

47. Какое увеличение ресурсов требуется, чтобы сократить сроки вдвое?

48. Сколько периодов имеется в процессе сверх тяжелого проекта?

49. Возможна ли организация работы с плотной укладкой по времени без смены ролевых функций исполнителей в случае выполнения только легких проектов? Обоснуйте ответ.

50. Какова при этом должна быть пропорция численности групп аналитиков и программистов? Аналитиков и эксплуатационников?

51. Как меняется доля работы аналитиков по сравнению с долей работы программистов с увеличением тяжести проекта?

52. Каковы этапы основной процедуры проведения типового проекта?

53. Какие основные задачи решаются на первом этапе процедуры проведения типового проекта? В каком порядке они должны выполняться?

54. Для каких типов проектов должно проводиться обследование? Какие методы при этом применяются?

55. Какова главная задача на первом этапе подготовки к проекту?

56. Каков порядок формирование бригады проекта?

57. На какие вопросы должен быть получен ответ при первичном анализе задачи?

58. Какие шаги выполняются в случае принятия решения о нецелесообразности продолжения подготовки к проекту?

59. На каком этапе рекомендуется принимать решение о назначении руководителя проекта?

60. Какие основные документы должны быть подготовлены утвержденным руководителем проекта?

61. Чем завершается этап подготовки к проекту?

62. Какие типы документов относятся к исходным материалам?

63. Являются ли Технические (Коммерческие) предложения обязательным документом? Обоснуйте ответ.

64. Является ли Техническое задание обязательным документом? Что определяет ТЗ?

65. Существуют ли способы организации работы над проектом при отсутствии технического задания или его эквивалента? Приведите примеры.

66. Какие документы входят в стандартный набор договорных материалов? Какие из них являются обязательными?

67. Определите алгоритм процесса согласования договорных материалов.

68. Какая модель процесса может быть использована на этапе работы над проектом?

69. От чего зависит состав промежуточных вех для каждой фазы проекта?

70. Какие вехи являются абсолютно обязательным для любой фазы?

71. От каких факторов зависит состав команды фазы проекта?

72. Процедура выполнения конкретной фазы проекта является предметом динамического или статического планирования? Обоснуйте ответ.

73. Каковы обязанности руководителя проекта на каждой фазе выполнения проекта?

74. Может ли руководитель играть определенные роли на различных фазах проекта?

75. Какую информацию должен содержать План выполнения фазы?

76. Какие средства используются обычно для составления Плана выполнения фазы?

77. Различаются ли требования, предъявляемые к качеству оформления внешних и внутренних материалов?

78. В чем состоит основное различие между внешними и внутренними материалами?

79. Основным результатом какой фазы является Концепция? Каково ее назначение? Как можно назвать Концепцию в соответствии с терминологией ЕСПД?

80. Что это такое «оценка риска»? На каком этапе производится начальная оценка риска?

810. Какие типы риска вы знаете?

82. На каких фазах производится переоценка риска?

83. Что является основным результатом фазы анализа?

84. Какие виды спецификаций вы знаете?

85. Какое основное средство рекомендуется использовать в качестве средства специфицирования?

86. Является ли ведение базы данных тестирования обязательным для любых проектов, связанных с разработкой программного кода?

87. Каково назначение третьего этапа основной процедуры проведения типового проекта?

88. Подписание акта сдачи-приемки работы является началом или завершением этого этапа?

89. Каким образом производится архивирование результатов работы?

90. Какие сведения должны входить в отчет о результатах проекта?

91. Какие приложения к отчету должен составить руководитель проекта?

92. Перечислите основные документированные процедуры, которые описывают порядок применения модели безотносительно к проектам?

93. В чем назначение процедуры «Учет рабочего времени»? Для чего используются собранные данные?

94. В чем назначение процедуры «Периодическая отчетность»? Какие виды и формы отчетности вы знаете?

95. Для чего нужна проверка качества материалов? Кто ее осуществляет?

96. Перечислите основные положения, которые должны неукоснительно выполняться при проведении любого проекта.

97. Нужно ли согласовывать с заказчиком результаты вех? Вовлекать заказчика в практическое использование результатов проекта на ранней фазе опытной эксплуатации?

Рекомендуемая литература по теме

1. Бек К. Экстремальное программирование. – Спб.: Питер, 2002.

2. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.

3. Брукс-мл. Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1975; новое издание перевода: Мифический человеко-месяц. СПб.: СИМВОЛ+, 1999.

4. Орлов С. Технологии разработки программного обеспечения. – СПб.: Питер, 2002.

5. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

6. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб : Питер, 2002.

Тема 4. Модели команды разработчиков

В современных условиях программное обеспечение разрабатывается коллективами, иногда очень большими коллективами. В таких коллективах проблемы взаимодействия являются постоянными и иногда очень сложными. Сами проблемы, несомненно, относятся к процессу разработки программного обеспечения, а способы решения этих проблем относятся к технологии программирования.

В рамках этой темы принимается как постулат предположение, что разработка программного обеспечения ведется некоторой группой людей, которая называется командой проекта. Команда проекта может иметь различные свойства. Команда может быть большой или маленькой, может иметь постоянный или переменный состав по ходу проекта, может быть постоянной или временной, созданной для одного проекта.

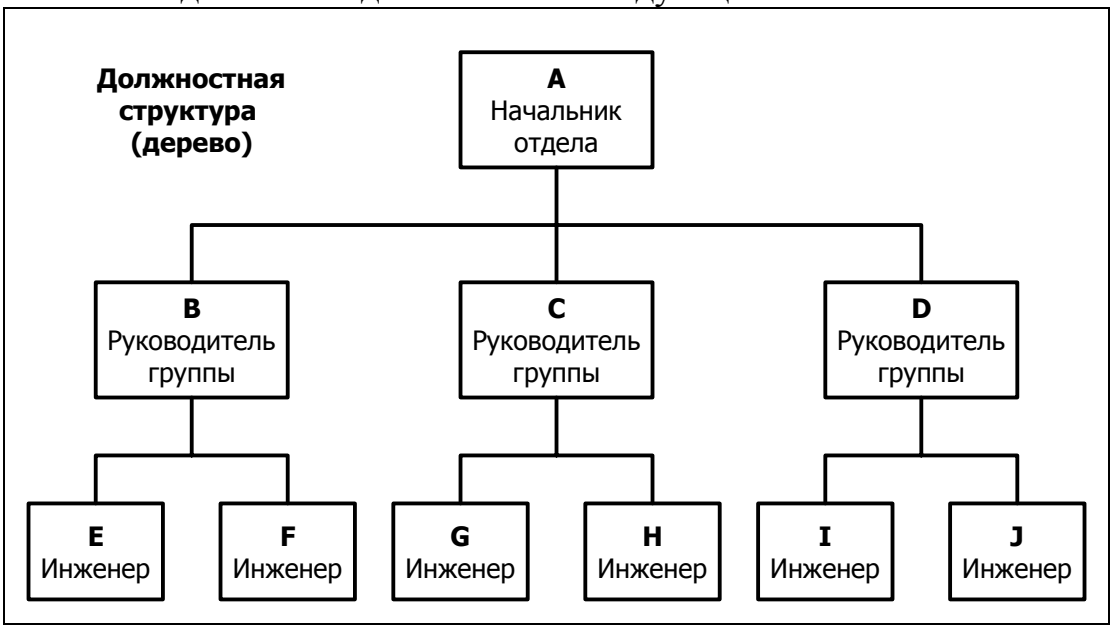
В четвертой теме рассматриваются следующие вопросы.

- 4.1. Коллективный характер разработки
  - 4.1.1. Оптимальный размер команды
  - 4.1.2. Подчиненность участников проекта
  - 4.1.3. Развитие команды и развитие персонала
  - 4.1.4. Специализация, кооперация и взаимодействие
- 4.2. Иерархическая модель команды
- 4.3. Метод хирургической бригады
- 4.4. Модель команды равных
- 4.5. Конструирование модели команды
  - 4.5.1. Особенности организации и требования к команде
  - 4.5.2. Архитектура модели команды
  - 4.5.3. Функции, роли и должности
  - 4.5.4. Статико-динамическая структура команды
  - 4.5.5. Распределение полномочий и ответственности
  - 4.5.6. Достоинства и недостатки модели команды

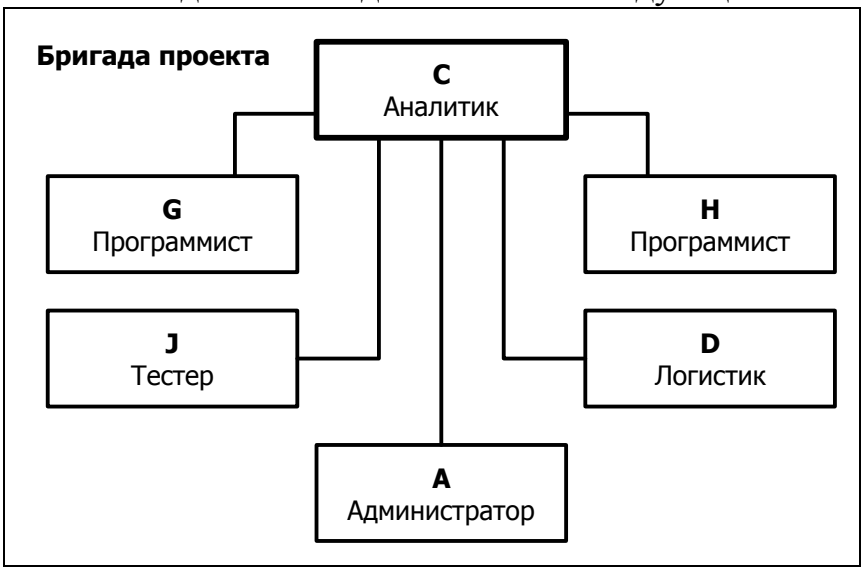
Вопросы для самопроверки

1. Что такое «модель команды»?
2. Что такое «команда проекта»? Сколько человек должно входить в команду проекта? Постоянен ли ее состав?
3. Какие основные виды организации подчиненности вы знаете?
4. Что такое линейно-функциональная организация? Каковы ее достоинства и недостатки
5. Что такое проектно-ориентированная организация? Каков ее основной недостаток?
6. Что такое матричная структура организации? Каковы ее основные достоинства и недостатки?
7. По каким направлениям происходит развитие команды в результате совместной работы над проектом?
8. Что такое специализация? Полезна ли специализация для успеха работы команды?
9. Что такое кооперация и взаимодействие?
10. Какая модель команды является самой распространенной?
11. Что такое «принцип единоначалия»? Чем он хорош и когда используется?
12. Требует ли использование иерархической модели дополнительных мероприятий по внедрению и обучению персонала?
13. Применима ли иерархическая модель, когда число исполнителей превышает миллион исполнителей?
14. Какие недостатки иерархической модели вы знаете?
15. Имеют ли значение личные качества руководителей при использовании иерархической модели?
16. Можно ли отнести иерархическую модель к экстенсивным моделям?
17. Влечет ли использование иерархической модели жесткое закрепление за каждым исполнителем его ролевой функции?
18. Чем отличается модель главного программиста от метода хирургической бригады?

19. Какие достоинства имеет модель главного программиста?
20. Какие модификации допускает модель звезды?
21. При использовании модели главного программиста может ли быть два Главных программиста?
22. Перечислите недостатки модели главного программиста.
23. Можно ли разделить крупный проект на несколько частей и запустить несколько бригад главного программиста параллельно?
24. Применим ли метод хирургической бригады, когда число исполнителей превышает тысячу исполнителей?
25. Чем характеризуется модель команды равных?
26. Как можно схематически изобразить модель команды равных?
27. Каковы преимущества и недостатки модели команды равных?
28. Применима ли модель команды равных, когда число велико, более ста человек?
29. Можно ли на практике использовать комбинации различных моделей?
30. К какой модели команды относится следующая схема?



31. К какой модели команды относится следующая схема?



32. Какие факторы важно учитывать при конструировании модели команды?

33. Какие основные функции предусматриваются при построении модели?

34. Что такое «Роль»? Может ли сотрудник, которому назначена определенная роль в рамках конкретного проекта, выполнять одновременно несколько функций?

35. Перечислите роли и функции, которые могут использоваться в проектах разных типов.

36. На каких фазах жизненного цикла приложения и какие функции выполняет аналитик, программист, тестировщик, эксплуатационник?

37. Что такое «должность»? Перечислите уровни иерархии должностей.

38. Что такое «должностная структура»? Является она статической или динамической? По какой модели обычно строится?

39. Является ли руководитель проекта административным начальником для всех вовлеченных в проект сотрудников?

40. Руководитель проекта – это функция, роль или должность?

41. Что такое «Бригада проекта»? Что определяется при назначении бригады? Приведите пример бригады проекта.

42. Что такое «Команда фазы»? Является она статической или динамической структурой? Приведите пример команды фазы.
43. Какой характер имеет распределение полномочий и ответственности в модели команды?
44. Кем назначается руководитель проекта, бригада проекта и команды фазы?
45. Опишите достоинства и недостатки предлагаемой модели команды.

Рекомендуемая литература по теме

1. Бек К. Экстремальное программирование. – Спб.: Питер, 2002.

2. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.

3. Брукс-мл. Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1975; новое издание перевода: Мифический человеко-месяц. СПб.: СИМВОЛ+, 1999.

Тема 5. Дисциплина программирования

В процессе разработки решающую роль играет индивидуальная продуктивность программистов. Индивидуальная продуктивность в немалой степени зависит от личных качеств программистов, но не только. Можно указать целый ряд факторов, не имеющих прямого отношения к личным качествами программистов, которые самым существенным образом влияют на индивидуальную продуктивность программирования. Именно такие факторы, а также приемы изменения их значений, составляют предмет дисциплины программирования.

Разнообразие технологических средств и приемов, применяемых программистами, поистине необозримо. Указать среди них два-три «наилучших в большинстве случаев» практически невозможно.

Поэтому в данной теме приведены общие соображения, относящиеся к дисциплине программирования, и указаны *некоторые* средства и приемы дисциплины программирования. Упоминаемые средства и приемы имеют ограниченную область применения: в некоторых случаях они дают выигрыш в продуктивности программирования, а в других случаях оказываются бесполезными. Таким образом, необходимо указать область применимости указываемых средств.

Природа программирования (в рассматриваемом типичном случае) содержит имманентно присущие ей противоречия, которые являются причиной основных проблем организации программирования:

С одной стороны, в распоряжении программиста находится код программы, который существенно конечен и существенно статичен и на который программист может воздействовать произвольным образом. С другой стороны, целью программирования является получение разворачивающегося во времени процесса выполнения программы, причем этот процесс потенциально бесконечен (или необозримо велик), существенно динамичен и не подлежит прямому управлению со стороны программиста. Таким образом, программирование по существу является процессом косвенного отложенного управления (планирования), причем цепочки прямых и обратных связей очень длинны и запутаны в современных программно-аппаратных компьютерных системах. Коротко говоря, программист работает с конечным статическим текстом программы, а представляющим интерес результатом является бесконечный динамический процесс выполнения,

и прямое соответствие между этими сущностями установить трудно или невозможно. Это противоречие между статикой и динамикой.

С одной стороны, в распоряжении программиста находится неограниченное количество экземпляров атомарных и композитных объектов, которые он может комбинировать произвольным образом, конструируя программу. Таким образом, качественных ограничений на возможные комбинации, характерных для реального мира, в идеальном мире программных объектов не существует. Это порождает эффект "комбинаторного взрыва", с которым программисту справиться не просто, поскольку количественные возможности комбинаторного мышления человека чрезвычайно ограничены. С другой стороны, компьютер способен выполнить программу неограниченного объема и сложности — количественных ограничений практически не существует. Более того, интерес представляют именно большие программы, которые программист не может в полном объеме единовременно обозреть и не может выполнить вручную. В то же время компьютер качественно ограничен в своих возможностях выполнения программы — выполнение носит строго детерминированный характер. Коротко говоря, цель программирования (выполнение программы) превосходит по объему непосредственные количественные возможности программиста. Это противоречие между количеством и качеством.

Фактически известен только один метод разрешения этих противоречий — принцип "разделяй и властвуй". Суть этого метода применительно к программированию состоит в наложении на программу внешней структуры, которая бы обеспечивала отслеживание соответствия между статикой и динамикой и позволяла бы справиться с количеством рассматриваемых комбинаций.

В программирующей организации дисциплина программирования является частью корпоративного стандарта и в этом смысле является нормативным документом. В тоже время, дисциплина программирования нацелена на непрерывное улучшение и совершенствование программирования в организации, а не на фиксацию и замораживание существующего порядка. Поэтому положения дисциплины программирования являются, в основном, рекомендующими и предписывающими, а не ограничивающими и запрещающими.

Несмотря на гибкость дисциплины программирования, в ней имеется неизменяемое жесткое ядро. Далее следует список основных положений, которые являются характеристическими для дисциплины программирования и которые должны неукоснительно выполняться при проведении каждого проекта.

- Программирование рассматривается как экономически обоснованная деятельность, имеющая целью получение прибыли организацией.
- Дисциплина программирования ориентирована на повышение продуктивности программирования за счет снижения доли внепланово изменяемого кода и увеличение доли повторно используемого кода.
- Модель жизненного цикла программы в дисциплине программирования согласована с моделью процесса.
- Повышение продуктивности в дисциплине программирования достигается за счет унификации уровня представления информации в циклах повышения продуктивности.
- Дисциплина программирования постулирует примат проектирования над кодированием, тестированием и отладкой.
- Проектирование разделяется на три стадии: концептуальное, логическое и детальное проектирование, которые выполняются, соответственно на фазах анализа, проектирования и реализации. Во всех случаях конкретному кодированию предшествует абстрактное моделирование.
- Кодирование в дисциплине программирования следует общепринятым правилам хорошего стиля и рассматривается как публичная деятельность, а код является корпоративным достоянием.
- Тестирование в дисциплине программирования является планируемым и протоколируемым.

Продуктивность программирования является мерой умелости программистов, а потому дисциплина программирования является сознательной дисциплиной.

В пятой теме рассматриваются следующие вопросы.

5.1. Природа программирования

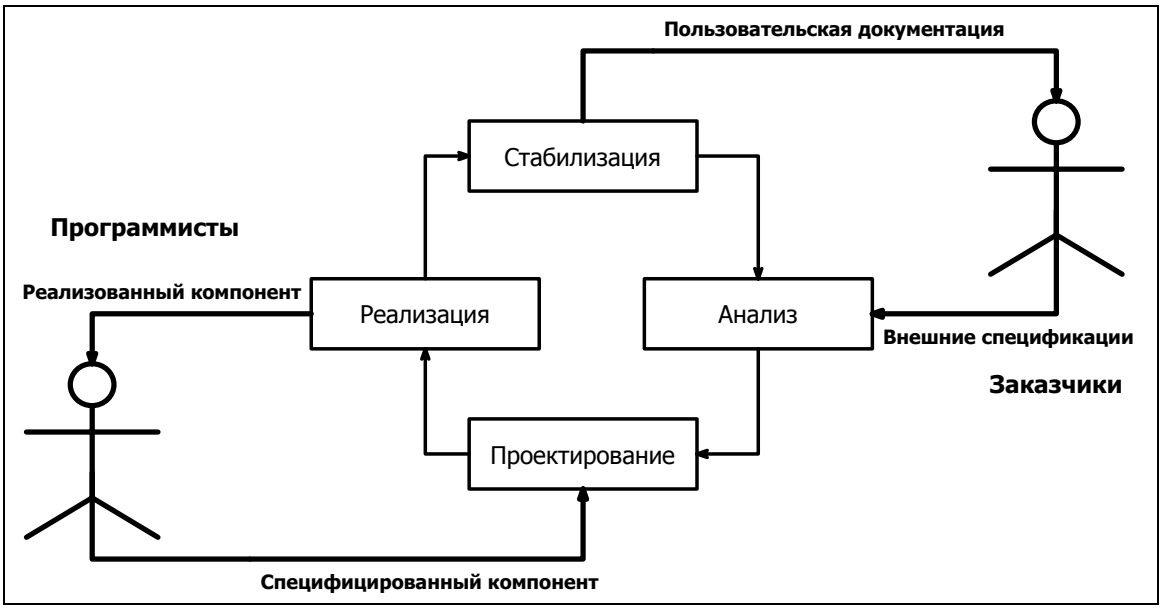
5.1.1. Наука программирования

5.1.2. Искусство программирования

Вопросы для самопроверки

1. Что такое «дисциплина программирования»?
2. Какие противоречия присущи природе программирования?
3. Является ли программирование наукой, искусством или ремеслом?
4. Можно ли сказать, что программирование в настоящее время носит ремесленный характер? Почему?
5. Что является предметом технологии программирования?
6. Что такое «парадигма программирования»?
7. Какие парадигмы программирования вы знаете?
8. В чем разница между концепцией структурного программирования и программированием без **go to**?
9. Какие структуры управления являются базовыми в концепции структурного программирования?
10. Каким свойством обладают все эти структуры?
11. Что такое «метод пошагового уточнения»?
12. Что такое «модуль»? Как называются модули в различных системах программирования?
13. Что такое «Программирование сверху вниз»?
14. Что такое «Программирование снизу вверх»?
15. Возможно ли применение комбинации этих подходов?
16. Что такое «Программирование вширь»?
17. Что такое «Модель вычислимости»?
18. Какие модели вычислимости вы знаете?
19. В какой архитектуре компьютеров отражена модель вычислимости, представленная машиной Тьюринга?
20. Какие фундаментальные понятия индуцируются такой архитектурой компьютера?
21. Как называется программирование, в котором используются эти ключевые понятия?
22. Что такое «Процедурное программирование»?

23. Что такое «Непроцедурное программирование»?
24. Что такое «Логическое программирование»? Является логическое программирование императивным или декларативным?
25. На каком методе основан язык Пролог?
26. Что такое «Продукционное программирование»?
27. В какой сфере часто применяются продукционные программы?
28. Что такое «Функциональное программирование»?
29. Существуют ли в промышленных масштабах компьютеры нетрадиционной архитектуры, для которых непроцедурное программирование является более естественным или это дело будущего?
30. Что является центральной идеей парадигмы объектно-ориентированного программирования? Какие еще понятия ассоциируются с ООП?
31. В каких системах программирования можно создавать ОО программы?
32. Каковы задачи системных аналитиков?
33. На что нацелена дисциплина программирования?
34. За счет каких факторов возможно повышение продуктивности программирования?
35. От чего зависит значение этих факторов?
36. Назовите и охарактеризуйте циклы движения информации на следующей схеме



37. Приведите пример классификации типов ошибок.
38. Ошибки какого типа снижают продуктивность в наибольшей степени?
39. Какие компоненты допускают повторное использование и реально повторно используются на практике?
40. Что влияет на выбор уровня формализации информации, циркулирующий в циклах повышения продуктивности?
41. Какие типы информации можно выделить в циклах повышения продуктивности?
42. Что такое «Спецификация»? Какие требования выдвигаются на первый план для разных видов спецификаций?
43. Какое средство спецификации наиболее целесообразно использовать, чтобы одновременно удовлетворить этим противоречивым требованиям?
44. Что такое «Архитектура приложения»?
45. Какие факторы оказывают влияние на архитектуру приложения?
46. Может ли существовать типовая архитектура (или модель) приложения?
47. Сколько возможно архитектурных решений интерфейса с пользователем?

48. Что такое «Активный интерфейс» и «Пассивный интерфейс»?
49. Какой тип интерфейса является более распространенным в настоящее время?
50. Какими причинами это обусловлено?
51. Что такое «Событийное управление»? Что такое «Реакция на событие»?
52. Есть ли в программе, целиком управляемой событиями, основной поток управления?
53. Что такое «Мастер»? Какими средствами он обычно реализуется?
54. Что такое «Архитектура клиент/сервер»? Когда этот термин стал активно использоваться?
55. Какие способы реализации архитектуры клиент/сервер вы знаете?
56. Что такое «Служба»?
57. От чего зависят методы реализации служб? Приведите примеры способов реализации служб.
58. В чем отличие оперативной службы от постоянной службы?
59. Что такое «Трехслойная архитектура»? Каковы ее достоинства?
60. Что такое «Проектирование программ»? Какие виды проектирования вы знаете?
61. На какой фазе выполняется концептуальное проектирование?
62. Какие цели должны быть достигнуты на стадии концептуального проектирования?
63. Что является основным выходом стадии концептуального проектирования?
64. Передаются ли заказчику результаты концептуального проектирования?
65. На какой фазе выполняется логическое проектирование?
66. Что является основным содержанием и результатом логического проектирования?
67. Задачей какого вида проектирования трехслойной модели приложения масштаба предприятия является вычленение и спецификация служб?

68. На какой фазе выполняется детальное проектирование? В чем оно заключается?
69. Какие элементы проекта создаются на стадии детального проектирования?
70. Какие вопросы должны быть решены для проектирования компонентов?
71. Приведите пример упаковки компонентов в службы.
72. Что такое «Скелет кода»? Поддерживается ли автоматическая генерация кода скелета?
73. Существует ли жесткие границы между стадиями проектирования?
74. Как называется стандартный язык запросов реляционных СУБД?
75. Что такое «Схема базы данных»?
76. Что такое «Нормальная форма»?
77. В каких случаях реляционные СУБД хорошо приспособлены для работы?
78. На какие характеристики приложения существенным образом влияет качество решения задачи проектирования схемы БД?
79. Целесообразно ли использование дополнительных инструментов для проектирования схем БД?
80. Возможны ли средства автоматизации построения графического интерфейса пользователя?
81. Что такое «Прототип интерфейса»?
82. Целесообразно ли использование средств автоматического построения прототипа интерфейса?
83. Какие важные характеристики позволяют обеспечить такие средства?
84. Что такое «Согласованность интерфейса»?
85. На какой стадии проектирования целесообразно проводить построение прототипа интерфейса?
86. Что такое «Кодирование»? На какой фазе оно выполняется?
87. Что такое «Программирование по образцу»? Приведите примеры форм представления и использования образцов кода.

88. Что такое «Образец проектирования»? Из каких основных элементов обычно состоит описание образца?
89. Разрешима ли задача построения доказательства правильности любой программы?
90. Для каких целей целесообразно использовать ручное доказательство правильности программ?
91. Назовите три основные формы вставки утверждений в программу.
92. Назовите приемы повышения читабельности программы.
93. Что такое «Комментарий»?
94. Какие аспекты должна отражать дисциплина имен?
95. Что такое «Венгерская нотация»? В чем ее достоинства и недостатки?
96. Какие вы знаете приемы расположение текста программы, улучшающие ее читабельность?
97. Что такое «Тестирование» и «Отладка»?
98. ПО каким основным направлениям шло развитие технологии программирования, ориентированное на повышение продуктивности?
99. Возможно ли исчерпывающее тестирование нетривиальной программы?
100. Что такое «Представительный набор тестов»?
101. Что такое «Регрессионное тестирование»? Когда целесообразно его применять?
102. Что такое «Тестирование по соглашению»?
103. Какие виды комплексного тестирования вы знаете?
104. Какие типичные приемы применяются при тестировании устойчивости?
105. Какие параметры проверяются при тестировании применимости?
106. Является ли обязательным требованием дисциплины программирования о наличии плана тестирования, регламентирующего порядок проведения тестирования по видам тестов и критериям приемлемости? Обоснуйте ответ

107. В чем заключается отладка?
108. Что является более трудоемким: локализация ошибок или их исправление?
109. Каков самый распространенный прием локализации ошибок?
110. Чем удобен интерактивный отладчик?
111. Что такое «Трассировка»?
112. Что такое «Отладочный стенд»?
113. Что такое «Структурная экспертиза»?
114. Какие формы может иметь исправление ошибок?
115. Является ли выбор инструментальных средств личным делом участника проекта?
116. Должны ли все разработчики использовать один и тот же инструмент из данного класса инструментов?
117. Перечислите характеристики дисциплины программирования.

Рекомендуемая литература по теме

1. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. – Спб.: Питер, 2004.

2. Бек К. Экстремальное программирование. – Спб.: Питер, 2002.

3. Вирт Н. Алгоритмы + структуры данных = программы. М., Мир, 1978.

3. Вирт Н. Систематическое программирование. Введение. М.: Мир, 1977.

4. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.

5. Дейкстра Э. Дисциплина программирования. М.: Мир, 1978.