

Вопросы для самопроверки

Программирование на языке C

11 ноября 2014 г.

- *Лекция 1. Вводная*

1. Какие существуют аспекты разработки ПО? Подходы к разработке?
2. Какие признаки характеризуют любительский подход?
3. Какие этапы входят в жизненный цикл ПО?
4. Как называется самая известная серия рисунков, посвященная ошибкам проектирования?
5. Какие события повлияли на возникновение и распространение языка C?
6. Кто является автором языка C?
7. Как называлась первая книга о языке C?
8. Как можно отследить динамику интереса к языкам программирования?
9. Что можно сказать о стандартах языка C?
10. Какие принципы характеризуют язык C?
11. Какие достоинства у языка C?
12. Какие недостатки свойственны C?
13. Что входит в инструментальные средства C?
14. Как происходит процесс построения программы?
15. Как традиционно называется первая программа? Что она делает?
16. Как влияет неправильное форматирование на восприятие текста программы?
17. К чему может привести неправильное форматирование текста программы?

18. Как построить и запустить программу в ОС Unix?
19. С какими параметрами вызывается gcc?
20. Какие существуют виды интерфейсов?
21. В чём достоинства командного интерфейса?
22. В чём достоинства графического интерфейса?
23. Какие существуют известные библиотеки для построения графических интерфейсов?
24. Что предлагает стандарт C99?
25. Что появилось в стандарте C11?

• *Лекция 2. Организация данных*

1. Опишите роль данных при работе программы.
2. Как данные могут поступать в программу?
3. Как связаны между собой данные и память?
4. Как связаны между собой данные и типы?
5. Что такое виртуальная машина для языка Си?
6. Что представляет собой простейшая модель памяти?
7. Как происходит распределение памяти? Перечислите виды распределений.
8. Перечислите основные области памяти. Что для них характерно?
9. Как можно классифицировать типы данных?
10. Как объявить переменную?
11. Как объявить несколько одноименных переменных?
12. Как объявить локальную переменную?
13. Как объявить файловую переменную?
14. Что указывается при инициализации? При фиксации?
15. Какие атрибуты есть у переменной?
16. Как объявить одномерный массив?
17. Как инициализировать одномерный массив?
18. Как объявить многомерный массив? Как его инициализировать?

19. Как можно представить текст в программе?
20. Какими способами можно задать значение символьной переменной?
21. Какие символы мы относим к специальным? Как они задаются?
22. Что такое строка?
23. Какими способами можно задать строку в программе?
24. Какие устройства ввода/вывода считаются стандартными?
25. Что представляет собой терминал?
26. Перечислите стандартные потоки ввода/вывода.
27. Перечислите функции потокового ввода/вывода.
28. Перечислите функции стандартного ввода/вывода.
29. Как локализовать программу?
30. Как быстро оценить значение большого числа?
31. Опишите принцип формирования числа в позиционной системе счисления.
32. Перечислите наиболее распространенные системы счисления.
33. Как перевести число из двоичной системы в десятичную?
34. Как перевести число из десятичной системы в двоичную?
35. Что такое build-time? Run-time?
36. Что такое объявление? Инициализация?
37. Что такое фиксация?
38. В чем состоит назначение оболочки времени выполнения?

• *Лекция 3. Операторы и выражения*

1. Какие уровни образуют архитектуру программы на C?
2. Что такое оператор? Выражение?
3. Чем оператор отличается от выражения?
4. Какие бывают операторы?
5. Приведите примеры операторов и выражений.
6. Приведите пример классификации операций.
7. Перечислите основные характеристики операций.

8. Чем левоассоциативные операции отличаются от правоассоциативных?
9. Как называются операции в зависимости от количества аргументов?
10. Какие операции относятся к высоко приоритетным? К низко приоритетным?
11. Что представляет собой истинное значение в Си? Ложное значение?
12. Чем отличается l-value-выражение от r-value?
13. В чём заключаются особенности присваивания в языке Си?
14. Как осуществляется преобразование типов?
15. Что такое явное и неявное преобразование типов?
16. Как работает механизм логических операций?
17. Какие бывают операции инкремента/декремента?
18. Как можно выразить операцию инкремента в разных формах?
19. Как вычисляется выражение с инкрементом в префиксной форме?
20. Как вычисляется выражение с инкрементом в постфиксной форме?
21. Какими особенностями обладают операции инкремента/декремента?
22. Что означает термин "неопределённое поведение"?
23. Какие алгоритмические конструкции мы относим к основным при структурном программировании?
24. Как можно классифицировать операторы?
25. Опишите варианты оператора if. Для чего используется этот оператор?
26. На что влияет неудачно выбранная конструкция оператора?
27. Какой должна быть сложность конструкций языка C?
28. Опишите формат оператора switch.
29. В чём назначение оператора switch?
30. Опишите формат оператора while. Для чего используется этот оператор?
31. Как можно использовать while при обработке строк?

32. Какие ошибки могут возникать при использовании `while`?
33. Опишите формат оператора `do..while`.
34. Чем цикл `do while` отличается от цикла `while`?
35. Опишите формат оператора `for`.
36. Чем оператор `for` отличается от других операторов цикла?
37. Как с помощью `for` инвертировать символы в строке?
38. Перечислите операторы управления ходом выполнения программы.
39. Что такое спагетти-код?
40. Что такое точки следования?

● *Лекция 4. Указатели*

1. Какие бывают способы адресации?
2. В чем состоит прямой доступ к переменной?
3. Что такое косвенная адресация?
4. Как объявить и использовать указатель на указатель?
5. Для чего используются указатели?
6. В чем состоят особенности работы с указателями?
7. На что объявить указатель?
8. Как объявить и инициализировать указатель?
9. В чем опасность непроинициализированных указателей?
10. Как работает инкремент значения указателя?
11. Как связать указатель и массив?
12. Что представляет из себя массив указателей?
13. В каком смысле говорят о взаимозаменяемости массивов и указателей?
14. Чем отличается строка, заданная с помощью указателя от строки, заданной с помощью массива?
15. Что такое указатель на константу? Как его объявить?
16. Что такое константный указатель? Как его записать?
17. Что такое константный указатель на константу? Как его использовать?

18. Как объявить указатель на массив?
19. Чем отличается указатель на массив от массива указателей?
20. Где используются указатели на массивы?
21. Как объявить и использовать указатель на функцию?

- *Лекция 5. Функции*

1. Что такое функциональная декомпозиция?
2. Как описывает программу функциональная модель?
3. Что такое многопоточная программа?
4. Какие операции характерны для функции?
5. Что такое описание функции?
6. Что такое прототип?
7. Какие классы памяти могут использоваться при описании функции?
8. Что возвращает функция, если не указывать явно тип возвращаемого значения?
9. Как организуются функции в одно-файловой программе?
10. Как организовать многофайловую программу с функциями?
11. В чем отличие между фактическими и формальными параметрами функции?
12. Почему опасно полагаться на последовательность обработки параметров функции?
13. Как параметр передается по значению? Когда используется такой способ передачи параметров?
14. Как передается параметр через указатель? Когда используется такой способ передачи параметров?
15. Как осуществляется передача массива в функцию?
16. Как передать в функцию двумерный массив?
17. Как запустить программу с параметрами?
18. Как могут поступать данные в программу?
19. Как описывается заголовок `main` при передаче ей параметров из командной строки?
20. Что означают параметры `main`?

21. Что такое рекурсивная функция?
22. Какие существуют разновидности рекурсии?
23. Опишите алгоритм построения рекурсивной функции?
24. Как рекурсивно реализовать вычисление факториала? Чисел Фибоначчи?
25. Что такое линейный рекурсивный процесс?
26. Чем опасен ЛРП?
27. Как образуется ряд Фибоначчи?
28. Как можно реализовать процедуру получения ряда Фибоначчи?
29. Что такое древовидная рекурсия?
30. Чем опасна древовидная рекурсия?
31. Как можно представить древовидный рекурсивный процесс?
32. Что такое хвостовая рекурсия?
33. Чем полезна хвостовая рекурсия?

• *Лекция 6. Пользовательские типы данных*

1. Как классифицируются стандартные и нестандартные типы в С?
2. Что такое псевдонимы (typedef)?
3. Для чего используются псевдонимы?
4. Как с помощью typedef упростить сложное объявление?
5. Как задать перечисление?
6. Для чего используются перечисления?
7. Как связать перечисление со строкой?
8. Что такое структура (запись)?
9. Чем запись отличается от массива?
10. Что строят на основе фундаментальных способов организации данных?
11. Как определить структуру?
12. Как проинициализировать структуру?
13. Перечислите свойства структурного типа.

14. Чем структуры отличаются от массивов?
15. Как в структуре осуществляется доступ к значениям полей?
16. Как создать вложенные структуры?
17. Можно ли присваивать структуры?
18. Как передать структуру в функцию?
19. Можно ли вернуть структуру из функции?
20. Могут ли указатели выступать в роли полей структур?
21. Может ли структура содержать указатель на себя саму?
22. Как задать объединение?
23. Чем объединение отличается от структуры?
24. Чему равен размер памяти, занимаемый структурой?
25. Чему равен размер памяти, занимаемый объединением?
26. Для чего используются безымянные объединения?
27. Для чего используются объединения?
28. Что такое поле битов?
29. Что происходит, если суммарный размер полей превышает 32 бита?
30. Приведите пример битового поля с "дырами".

● *Лекция 7. Битовые поля и побитовые операции*

1. Опишите принцип формирования числа в позиционной системе счисления.
2. Перечислите наиболее распространенные системы счисления.
3. Как перевести число из двоичной системы в десятичную?
4. Как перевести число из десятичной системы в двоичную?
5. Как перевести шестнадцатеричное число в десятичное?
6. Как перевести шестнадцатеричное число в двоичное?
7. Как узнать, какой порядок байт в вашей системе?
8. Как можно представить байт на низком уровне?
9. Как можно перевести число в двоичную систему без деления?
10. Перечислите побитовые операции.
11. Чему равен результат поразрядного отрицания?

12. Как узнать максимальное значение целого беззнакового типа в системе?
13. Чему равен результат поразрядного И?
14. Чему равен результат поразрядного ИЛИ?
15. Чему равен результат поразрядного исключающего ИЛИ?
16. Как реализовать обмен двух целых без вспомогательной ячейки?
17. Что такое маска? Для чего могут использоваться маски?
18. Как включить и выключить специфические разряды?
19. Как можно переключить значение бита? Сравнить значение бита?

• *Лекция 8. Динамическая организация данных*

1. Перечислите свойства динамически выделенной памяти.
2. Какой общий порядок работы с динамической памятью?
3. Перечислите основные функции для работы с динамической памятью.
4. Как прочитать файл целиком в динамический массив?
5. Как создать динамический многомерный массив?
6. Как удалить многомерный массив?
7. Перечислите основные ошибки, связанные с динамической памятью.
8. Приведите пример ошибки, связанной с двойным освобождением динамической памяти.
9. Приведите пример утечки памяти.
10. Как может возникнуть утечка памяти при работе с многомерными массивами?
11. Какие способы организации данных можно отнести к фундаментальным?
12. Что строят на основе фундаментальных способов организации данных?
13. Какие достоинства и недостатки свойственны массивам?
14. В чём заключаются проблемы статических и динамических массивов?

15. Что такое связанный список? Чем он отличается от массива?
16. Какие бывают разновидности связанных списков?
17. В чем недостатки списков?
18. Как в С и С++ описываются элементы списка?
19. Как можно программно реализовать список?
20. Какие операции необходимы для работы со списком?
21. Что такое дерево?
22. Какую роль выполняет корень дерева?
23. Какую роль играет понятие поддеревы по отношению к дереву?
24. Что такое путь на дереве? Глубина дерева?
25. Как можно классифицировать деревья?
26. Что представляют из себя бинарные деревья?
27. Что такое вырожденное дерево?
28. Что такое полное дерево?
29. Какие операции считаются основными для бинарного дерева?
30. Из каких базовых действий строится алгоритм обхода дерева?
31. Чем отличаются друг от друга прямой, обратный и симметричный обходы дерева?
32. Что такое свойство упорядоченности?
33. Как осуществляется программная реализация бинарного дерева?
34. Что должен содержать узел бинарного дерева?
35. Какие операции над узлами дерева можно отнести к основным?

• *Лекция 9. Завершающие темы*

1. Что такое класс хранения?
2. Какие бывают классы хранения для переменных? Для функций?
3. Что такое область видимости?
4. Перечислите основные области видимости.

5. Что такое связывание? Какие бывают разновидности связывания?
6. Что такое длительность хранения? Какие разновидности длительности хранения бывают?
7. Что такое автоматический класс хранения?
8. Что такое экранирование переменных и когда оно возникает?
9. Когда нужно использовать класс хранения register?
10. К каким переменным может применяться static?
11. В чем разница между обычной (автоматической) переменной, объявленной внутри блока и static-переменной внутри блока?
12. Что происходит, если static применяется к файловой переменной?
13. Что можно сказать о видимости файловых переменных, объявленных с static?
14. Для чего используется класс хранения extern?
15. В чем заключается основная функция препроцессора?
16. Какие задачи решает препроцессор?
17. Перечислите основные директивы препроцессора.
18. Как включить текст одного файла внутрь другого?
19. Как создать макрос без параметров?
20. Как создать макрос с параметрами?
21. Какие ошибки могут возникнуть при использовании макросов с параметрами?
22. Что отличает многофайловые проекты от однофайловых?
23. С какими трудностями сталкивается разработчик больших однофайловых программ?
24. Перечислите правила построения многофайловых программ.
25. Что целесообразно включать в заголовочные файлы?
26. Что нецелесообразно включать в заголовочные файлы?
27. Что нельзя включать в заголовочные файлы?
28. Как правильно организовать заголовочный файл?