

Програмни Езици

(обектно –ориентирано програмиране)

1. Обекти и класове Дефиниция на клас. Общи понятия и концепции.
2. Методи и параметри. Даннови членове. Инстанция на обект.
3. Класове и обекти. Преход от структура към клас.
4. Конструктори и деструктори
5. Типове променливи и оператори в клас. Основни типове данни.
6. Указатели и референции. Константи. Изброими типове.
7. Предаване на данни като параметри. Accessor-методи. Mutator-метод.
8. Общи понятия: функции; декларация на прототип; дефиниране на функция; повикване на функция; работа с локални и глобални типове; дефиниране и използване на overloaded функции.
9. Функции –методи. Глобални функции
10. Припокриване на функции
11. Локални променливи. Локални и/или глобални елементи на клас.
12. Взаимодействия на обекти. Понятие за абстракция и модулност.
13. Програмни практики: цифров часовник: диаграми на класове и обекти; проект и програмни елементи.
14. Групиране на обекти. Колекции и итератори . Програмни практики: проект 'notebook'; обектна структура, използване на колекции.
15. Проектиране на класова йерархия в обектно-структурирана програма.
16. Оценка на качеството на кода. Съвързаност и структурираност на кода.
17. Качество на код: дублиращи се фрагменти. Целево-ориентиран проект.
18. Проблеми в проектиране и ООП. Водещи практики.
19. Класове и обекти: разделяне на декларация и дефиниция. Създаване и унищожаване на обекти.
20. Дефиниране на конструктори и деструктори. Методи на класа.
21. Дефиниране на връзки.
22. Наследяемост. Подтипове. Замествания. Полиморфизъм и променливи.
23. Програмни практики: разнородни обекти в бази от данни. Проект: обектен модел, класова диаграма, реализация. Подобрена версия с наследяемост. Суперклас и подкласове. Наследяемост.
24. Създаване на проекти с дълбока йерархия.
25. Подтипове, подкласове и присвоявания. Предаване на параметри.
26. Полиморфични променливи.
27. Наследяемост в ООП . Дефиниране на йерархията. Достъп до методи и данни на различни нива.
28. Виртуални елементи и реализация на полиморфизма. Абстрактни класове и абстрактни методи.
29. Програмни практики: наследяемост и полиморфизъм в програма.
30. Виртуални функции викане на виртуални функции на базов клас.
31. Вграждане на обекти. Сору – конструктори.
32. Присвоявания и обекти.
33. Референтни параметри (const или non-const). Работа с референции.
34. Връщане на референтни обръщания.
35. Конструирание на вградени обекти. Деструкция на вградени обекти.
36. Заделяне на обекти от динамичната памет. Проблеми, породени от взаимодействията между обекти.
37. Виртуални деструктори.

38. Приятелски класове и приятелски функции.
39. Софтуерни контракти. Пред-условия, пост-условия и инварианти към методи на клас. Синтаксис в .NET. Примери.
40. Контракт за инварианти – поглед в дълбочина. Реализация в .NET.
41. Контракт и наследяване. Проблем с ограничаване на областта в дъщерен обект.
42. Статични членове на клас.
43. Припокриване на оператори. Същност, ограничения. Реализация.
44. Програмни практики: припокриване на аритметични операции.
45. Преобразувания и операции -преобразувания.
46. Пример - Microsoft Foundation Class Library (MFC).
47. Обектен дизайн : принципи на SOLID, open/closed принцип, принцип на регламентираната отговорност.
48. Обектен дизайн : принцип на верижната отговорност,
49. Обектен дизайн: принцип на двойния dispatch (пренасочване) в run-time.
50. Обектен дизайн : принцип на Лисков. Принцип на Лисков и контрактите в .NET.
51. Нови стандартизирани конструкции в C++. Асинхронни задачи, нишки и локални за тях данни.
52. Конструктори в C++ за синхронизация: атомични типове и условни променливи, мютекси и заключвания.
53. Генетични (пораждащи)типове в обектното програмиране. Синтаксис. Начин на обработка в .NET среда. Разлика с шаблонизирани типове.
54. Генетични типове и наследяемост. Синтактично подменяне на генетичен тип. Обработка на генетични типове. Ограничители.
55. Съвременни езикови конструктори: lambda expression; функции-обекти. Сравнение, анализ.
56. Обекти в паметта – особености при разполагането и чести програмни грешки
57. Управление на памет в конзолен режим и в Linux системи: служебни структури в паметта. Начин на работа на системата за освобождаване (макрос unlink()).
58. Препълване на буфер. Поглед отвътре. Използване за недобросъвестно вмъкване на код. Пример.
59. Техниката 'frontlink' за сриване на код. Пример.
60. Опасности при двойно освобождаване на памет (double-free vulnerabilities). Пример.
61. Динамично управление на памет в Windows.
62. Служебни структури в динамичния мениджмънт на памет в Windows.
63. Препълване на буфер в Windows и атаки, базирани на това. Пример. Техники за вмъкване на код и пренасочване на управление.
64. Съпоставяне на файл с Оперативна Памет. Съвети при управление на паметта.

проф. д-р О. Наков