

Програмни Езици

1. Обекти и класове Дефиниция на клас. Общи понятия и концепции.
2. Методи и параметри. Даннови членове и пропъртита.
3. Модификатори на достъп в клас.
4. Accessor-методи. Mutator-метод.
5. Методи на клас. Видове и модификатори. Припокриване.
6. Програмни практики: цифров часовник: диаграми на класове и обекти; проект и програмни елементи.
7. Групиране на обекти.Колекции и итератори . Програмни практики: проект 'notebook'; обектна структура, използване на колекции.
8. Проектиране на класова йерархия в обектно-структурирана програма.
9. Оценка на качеството на кода. Свързаност и структурираност на кода.
10. Качество на код: дублиращи се фрагменти. Целево-ориентиран проект.
11. Проектиране на обектно-структуриран код.
12. Класове и обекти: разделяне на декларация и дефиниция. Създаване и унищожаване на обекти. Структура и обект.
13. Конструктори и деструктори. Видове конструктори (виртуални и статични). Методи на клас.
14. Дефиниране на връзки. Взаимодействия на обекти по вертикала и хоризонтала.
15. Подтипове, подкласове и присвоявания. Предаване на параметри.
16. Наследяемост. Полиморфизъм. Достъп до методи и данни на различни нива.
17. Виртуализация и реализация на полиморфизма. Абстрактни класове и абстрактни методи. Виртуални функции и викане на виртуални функции на базов клас.
18. Вграждане на обекти. Сору – конструктори. Присвоявания и обекти.
19. Референтни параметри (const или non-const). Работа с референции. Връщане на референтни обръщения.
20. Конструирване на вградени обекти. Деструкция на вградени обекти.
21. Заделяне на обекти от динамичната памет. Проблеми, породени от взаимодействията между обекти.
22. Приятелски класове и приятелски функции. Статични членове на клас.
23. Припокриване на оператори. Същност, ограничения. Припокриване на аритметични операции.
24. Преобразувания и операции -преобразувания.
25. CASE: обектно-структурирана програмна система - плащане. UML диаграми на класовете, дефиниране достъп, методи, пропъртита, права. Полиморфизъм при изпълнението и преобразувания над типове.
26. Разширяване функционалността на системата за плащания: въвеждане на интерфейси. Полиморфична употреба на методи, наследяване на интерфейс.
27. Често използвани интерфейси в .NET.
28. *Софтуерни контракти. Пред и пост-условия. Инварианти. Примери.
29. *Контракт за инварианти – поглед в дълбочина. Реализация в .NET.
30. *Контракт и наследяване. Проблем с ограничаване на областта в дъщерен обект.
31. Обектен дизайн : принципи на SOLID, open/closed принцип, принцип на регламентираната отговорност.

- 32. **Обектен дизайн : принцип на верижната отговорност,**
- 33. **Обектен дизайн: принцип на двойния dispatch (пренасочване) в run-time.**
- 34. **Обектен дизайн : принцип на Лисков. Принцип на Лисков и контрактите в .NET.**

- 35. **Генетични (пораждащи) типове в обектното програмиране. Синтаксис. Начин на обработка в .NET. Разлика с шаблонизирани типове.**
- 36. **Генетични типове и наследяемост. Синтактично подменяне на генетичен тип. Обработка на генетични типове. Ограничители.**

- 37. **Lambda- изрази. Синтаксис и използване на ламбда-изразите. Функции-обекти.**

- 38. **Обекти в паметта – особености при разполагането и програмни грешки.**
- 39. **Управление на памет в конзолен режим и в Linux системи: служебни структури в паметта. Освобождаване на обекти (макрос unlink()).**
- 40. **Препълване на буфер. Поглед отвътре. Техника на вмъкване на код. Пример.**
- 41. **Атаката от тип - 'frontlink'. Пример.**
- 42. **Опасности при двойно освобождаване на памет (double-free vulnerabilities). Пример.**
- 43. **Динамично управление на памет в Windows.**
- 44. **Служебни структури в динамичния мениджмънт на паметта в ОС Windows.**
- 45. **Препълване на буфер в Windows и атаки, базирани на това. Пример. Техники за вмъкване на код и пренасочване на управление.**
- 46. ***Съпоставяне на файл с оперативна памет. Програмни практики при управление на паметта.**