Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра информатики

А.А. Волосевич

ИНСТРУМЕНТЫ И СРЕДСТВА ПРОГРАММИРОВАНИЯ

Тестовые вопросы
для студентов специальности I-31 03 04 «Информатика»
всех форм обучения

Минск 2006

СОДЕРЖАНИЕ

ТЕСТ №1. ООП НА ЯЗЫКЕ OBJECT PASCAL 3

ТЕСТ №2. ЯЗЫК VISUAL BASIC.NET 5

ОТВЕТЫ НА ТЕСТ №1 8

ОТВЕТЫ НА ТЕСТ №2 10

ТЕСТ №1. ООП НА ЯЗЫКЕ OBJECT PASCAL

1. Если метод класса не работает с полями объекта, передается ли в метод параметр `self`?
2. В каком случае один класс может работать с `private`-данными другого класса?
3. Допустимо ли создавать класс-наследник на основе 2 родительских классов?
4. Можно ли в Object Pascal использовать перекрытие полей при наследовании?
5. Для чего используется ключевое слово `inherited`?
6. Что такое полиморфизм?
7. Как выглядит синтаксис объявления указателя на метод?
8. Можно ли объявить в секции `published` поле целого типа?
9. Можно ли в классе-наследнике изменить видимость поля с `private` на `protected`?
10. Можно ли перекрыть метод с изменением количества и типа параметров метода?
11. Если метод не работает с полями объекта, можно ли вызывать такой метод у объекта перед вызовом конструктора?
12. При вызове деструктора используется имя класса или имя объекта?
13. Какие классы содержат RTTI?
14. Можно ли, используя переменную типа метакласс, получить доступ к полям объекта?
15. Классовый метод получает параметр `self`?
16. Опишите внутреннюю структуру переменной типа указатель на метод.
17. Можно ли объявить свойство в секции `private`?
18. Можно ли в классе-потомке уменьшить видимость членов по сравнению с классом-предком?
19. Может ли класс содержать несколько свойств-массивов?
20. Указатель на VMT размещается в памяти перед полями объекта или после полей?
21. Можно ли объявить в одном классе несколько конструкторов?
22. В классе объявлены 3 поля. Может ли класс содержать объявления 5 свойств?

23. В классе объявлено поле целого типа `integer`, поле типа `Boolean`, свойство типа `word`, одна процедура и одна функция. Сколько байт динамической памяти будут занимать 100 объектов такого класса?
24. Можно ли описать класс в пользовательской процедуре?
25. В базовом классе объявлены 2 виртуальных метода, в производном классе добавлены еще 3 виртуальных метода. Сколько элементов будет содержать VMT производного класса?
26. Перечислите директивы ограничения видимости в порядке «увеличения открытости».
27. Вася пытается объявить в классе `TC` конструктор: `constructor Init(i: Integer): TC`; Почему данный код не компилируется?
28. Может ли тип индекса свойства-массива быть `string`?
29. Как должна быть объявлена объектная переменная, чтобы ей можно было присваивать объекты любых классов?
30. Можно ли объявить виртуальный конструктор?
31. Какое ключевое слово используется в Object Pascal для генерации исключительной ситуации?
32. Имеет ли значение порядок блоков обработки исключительной ситуации?
33. Компонент `TLabel` это наследник класса `TGraphicControl` или `TWinControl`?
34. В каком классе Object Pascal: `TPersistent` или `TComponent` – объявлен виртуальный конструктор?
35. В каком методе выполняется отрисовка компонента (в Object Pascal)?

ТЕСТ №2. ЯЗЫК VISUAL BASIC.NET

1. Опция `Option Explicit` может размещаться в любом месте исходного кода?
2. В операторе `Select Case` тестируемое выражение может быть строкой?
3. Как обнулить элементы массива, используя цикл `For Each`?
4. Может ли шаг цикла `For - Next` быть отрицательной дробной величиной?
5. Можно ли изменить размер массива во время работы программы в зависимости от данных, введенных пользователем?
6. Какое ключевое слово используется в VB.NET для указания того, что метод является перегруженным?
7. Разрешено ли преобразование `Double` в `Decimal` в режиме жесткой проверки типов?
8. Можно ли в одностроковой форме оператора условного перехода разместить после `Then` несколько команд?
9. Что является особенностью .NET Framework: a) автоматическая сборка мусора или b) запрет на явную работу с указателями?
10. Сколько байт занимает в VB.NET примитивный тип `Date`?
11. Опишите область видимости, определяемую модификатором `Friend`?
12. Если `A` и `B` – два массива одного типа, то присваивание `A = B` a) выполнить невозможно; b) приведет к поэлементному присваиванию массивов; c) приведет к присваиванию ссылок на массивы.
13. Можно ли выполнить выход из функции при помощи `Exit Function`, не установив перед этим возвращаемого значения функции?
14. Где можно присваивать значения полям, которые объявлены с модификатором `ReadOnly` – в конструкторе или в методе `Main()`?
15. Массивы в VB.NET: при объявлении указывается число элементов или верхняя граница индекса?
16. Какой модификатор доступа используется по умолчанию для методов класса – `Private`, `Friend` или `Public`?
17. Класс объявлен с модификатором `MustInherit`. Может ли такой класс содержать разделяемые методы?
18. Если делегат инкапсулирует функцию, можно ли несколько таких делегатов объединить в один групповой?
19. Какой оператор используется для генерации события?
20. Может ли интерфейс содержать разделяемые члены?

21. Могут ли два типа делегата, находящиеся в одном пространстве имен, иметь одно имя, но различаться своей сигнатурой?
22. Для того чтобы объявить групповой делегат, используется модификатор `Group`?
23. Может ли при описании функции в качестве типа возвращаемого значения быть указан интерфейс?
24. Можно ли в структуре объявить `Private`-конструктор без параметров?
25. Может ли перечисление содержать в качестве членов именованные константы строкового типа?
26. Могут ли в перечислении быть две константы с одним значением?
27. Может ли структура содержать в качестве своего члена событие?
28. Что верно: пользовательский тип модуль включает описание нескольких пространств имен или одно пространство имен может включать описание нескольких модулей.
29. Какое ключевое слово используется для генерации исключительной ситуации?
30. Могут ли пространства имен быть вложенными?
31. Может ли объявление класса содержать вложенное объявление другого класса?
32. Может ли интерфейс наследоваться от двух классов?
33. Может ли разделяемый конструктор класса принимать параметр целого типа?
34. Для чего используется ключевое слово `RemoveHandler`?
35. Можно ли свойство-массив сделать свойством по умолчанию, если тип его индекса – `String`?
36. Может ли класс одновременно реализовывать некий интерфейс и быть наследником некоего класса?
37. Можно ли описать в одном классе несколько свойств-массивов с разными типами индексов, но одинаковыми именами?
38. Может ли в классе быть объявлен пользовательский конструктор без параметров?
39. Можно ли объявить такой класс, который содержит единственный конструктор с модификатором доступа `Private`?
40. Может ли один класс реализовывать несколько интерфейсов, если эти интерфейсы содержат функции с одинаковыми именами?

41. Какой неабстрактный класс позволяет читать данные из потока как последовательность символов?
42. Верно ли, что `IDictionary` – наследник интерфейса `IList`?
43. Может ли быть применен метод `BeginInvoke()` к групповому делегату?
44. Можно ли запустить сборку мусора в программе принудительно?
45. Метод `Sort` класса `Array` – это разделяемый или экземплярный метод?
46. Как называется метод класса `BinaryWriter` для записи в поток значения переменной типа `System.Int32`?
47. Верно ли, что в случае одномерного массива его свойства `Rank` и `Length` всегда возвращают одинаковое значение?
48. Верно ли, что если вызвать `EndInvoke()` у делегата, асинхронный метод которого еще выполняется, то `EndInvoke()` вернет `Nothing` – пустую ссылку?
49. Сколько параметров и какого типа принимает функция `IComparable.CompareTo()`?
50. Класс `SortedList` реализует интерфейс `IDictionary` или `IList`?
51. Метод `P`, который имеет 2 `ByVal` и 3 `ByRef` параметра будет вызван асинхронно. Сколько параметров будет у метода `BeginInvoke()` соответствующего делегата, а сколько – у метода `EndInvoke()`?
52. Какой метод для освобождения ресурсов объекта можно вызвать вручную: `Finalize()` или `Dispose()`?
53. Пусть создан массив, у которого нижняя граница индекса равна 4. Можно ли для перебора элементов такого массива использовать цикл `For Each`?
54. Метод `GetInvocationList()` класса `System.MulticastDelegate` возвращает массив, элементами которого являются…
55. Может ли команда `Mid` быть записана в левой части оператора присваивания?

ОТВЕТЫ НА ТЕСТ №1

1. Да.
2. Если эти классы размещены в одном модуле.
3. Нет.
4. Да, но не рекомендуется.
5. Для вызова методов класса-предка.
6. Особый вид перекрытия методов при наследовании, при котором программный код, работавший с методами родительского класса, пригоден для работы с измененными методами дочернего класса.
7. `TMP = procedure . . . of object;`
8. Нет.
9. Нет.
10. Да.
11. Да.
12. Имя объекта.
13. Любые классы.
14. Нет.
15. Да.
16. 4 байта – указатель на объект, 4 байта – указатель на код метода.
17. Да.
18. Нет, нельзя.
19. Да.
20. Перед полями объекта.
21. Да.
22. Да.
23. 100* (4 + 4 + 1) = 900.
24. Нет.
25. 5 элементов.
26. `Private`, `protected`, `public`, `published`.
27. Конструктор – это не функция.
28. Да.
29. `var X : TObject`.

30. Да.
31. `raise`.
32. Да.
33. `TGraphicControl`.
34. `TComponent`.
35. Метод `Paint`.

ОТВЕТЫ НА ТЕСТ №2

1. Нет.
2. Да.
3. Никак нельзя.
4. Да.
5. Да.
6. `Overloads`.
7. Нет.
8. Да.
9. И a, и b.
10. 8 байт.
11. Элемент доступен без ограничений, но только в той сборке, где описан.
12. Вариант c.
13. Да.
14. В конструкторе.
15. Верхняя граница индекса.
16. `Friend`.
17. Да.
18. Да.
19. `RaiseEvent`.
20. Нет.
21. Нет.
22. Нет.
23. Да.
24. Нет.
25. Нет, только именованные целочисленные константы.
26. Да.
27. Да.
28. Второе: одно пространство имен может включать описание нескольких модулей.
29. `Throw`.
30. Да.

31. Да.
32. Нет; даже и от одного не может.
33. Нет, разделяемый конструктор – это всегда конструктор без параметров.
34. Для отключения обработчика события.
35. Да.
36. Да.
37. Да – перегрузка свойств.
38. Да.
39. Да (закрытый конструктор).
40. Да.
41. `StreamReader`.
42. Нет.
43. Да.
44. Да, используя `System.GC.Collect()`.
45. Это разделяемый метод.
46. `Write()`.
47. Нет, `Rank` – число измерений, `Length` – число элементов.
48. Нет.
49. Один, типа `Object`.
50. `IDictionary`.
51. 2+3+2 = 7 у `BeginInvoke()`, 3+1 = 4 у `EndInvoke()`.
52. `Dispose()`.
53. Да.
54. Делегаты, составляющие цепочку вызова.
55. Да.