

Темы к экзамену по дисциплине “Распределённые системы”.

Часть 1. .NET/C#.

1. ООП: инкапсуляция, наследование, полиморфизм.
2. Утиная типизация.
3. Модификаторы доступа public, private, protected, internal в языке C#.
4. Ключевые слова virtual, abstract, override, sealed в языке C#. Перегрузка функций.
5. Контексты применения ключевых слов new и this в языке C#.
6. Ключевые слова static, const и readonly в языке C#.
7. Ключевые слова in, ref, out в языке C# (в контексте передачи параметров в метод), различия и примеры использования.
8. Класс System.Object. Методы GetType, Equals, GetHashCode, ToString, ReferenceEquals, назначение. Правила и примеры переопределения.
9. Примитивные типы, типы значений и ссылочные типы.
10. Типы значений, допускающие значение null.
11. Обработка исключений. Классы System.Exception, System.ArgumentException, System.ArgumentNullException, System.IndexOutOfRangeException, System.AggregateException. Ключевые слова try, catch, finally, when, особенности использования. Раскрутка стека.
12. Поточный ввод-вывод. Классы System.IO.MemoryStream, System.IO.FileStream, System.IO.StreamReader, System.IO.StreamWriter, назначение, примеры использования.
13. Классы System.String и System.Text.StringBuilder. Особенности хранения строк в памяти. Интернирование строк. Структурный паттерн проектирования “Приспособленец”.
14. Сборка мусора. Особенности процесса сборки мусора. Принудительная сборка мусора. Поколения объектов. Куча больших объектов.
15. Сборка мусора. Деструктор и финализатор. Список на финализацию, очередь на финализацию. Метод GC.SuppressFinalize.
16. Сборка мусора. Интерфейс System.IDisposable. Декларация using.
17. Интерфейс System.ICloneable, особенности, реализация и использование (с примерами кода).

18. Интерфейсы `System.IEquatable<T>`, `System.Collections.Generic.IEqualityComparer<T>`, особенности, реализация и использование (с примерами кода).
19. Интерфейсы `System.IComparable`, `System.IComparable<T>`, `System.Collections.IComparer<T>`, особенности, реализация и использование (с примерами кода).
20. Интерфейсы `System.Collections.IEnumerator`, `System.Collections.IEnumerable`, `System.Collections.Generic.IEnumerator<T>`, `System.Collections.Generic.IEnumerable<T>`, особенности, реализация и использование (с примерами кода).
21. Явная реализация интерфейса, назначение, примеры.
22. Цикл `foreach` и когда его можно использовать. Именованные итераторы. Операторы `yield return` и `yield break`.
23. Делегаты. Классы `System.Delegate`, `System.MulticastDelegate`.
24. Делегаты `System.Action<...>`, `System.Func<...>`, `System.Predicate<T>`, `System.Collections.Generic.Comparer<T>`, `System.EventHandler`, `System.EventHandler<TEventArgs>`, назначение и примеры использования.
25. События (event), как они связаны с делегатами и чем отличаются от делегатов. Назначение и примеры использования.
26. Анонимные делегаты и лямбда-выражения, примеры использования.
27. Многопоточное программирование. Класс `System.Threading.Thread`, примеры использования. Делегаты `System.Threading.ThreadStart` и `System.Threading.ParameterizedThreadStart`, примеры использования.
28. Многопоточное программирование. Класс `System.Threading.ThreadPool`. Метод `QueueUserWorkItem` класса пула потоков, примеры использования. Класс `System.Threading.Tasks.Parallel`, назначение, примеры использования.
29. Асинхронное программирование. Когда необходимо использование асинхронного программирования. Различия между асинхронностью и многопоточностью.
30. Асинхронное программирование. Классы `System.Threading.Tasks.Task`, `System.Threading.Tasks.Task<T>`, структура `System.Threading.Tasks.ValueTask<T>`. Назначение и примеры использования.
31. Асинхронное программирование. Методы классов асинхронно выполняемых задач `ContinueWith`, `WaitAll/WaitAny`, `WhenAll/WhenAny`, `Run`, `ConfigureAwait`, примеры использования.
32. Асинхронное программирование. Класс `System.Threading.Tasks.TaskFactory`, назначение и связь с методом `Task.Run`.
33. Асинхронное программирование. Ключевые слова `async/await`, ограничения на использование, примеры использования.

34. Асинхронное программирование. Отмена асинхронно выполняемой задачи. Класс `System.Threading.CancellationTokenSource` и структура `System.Threading.CancellationToken`, назначение и примеры использования.
35. Пространство имён `System.Dynamic`. Классы `System.Dynamic.DynamicObject`, `System.Dynamic.ExpandoObject`, особенности и назначение.
36. Ключевое слово `dynamic`, связь с типом `System.Object`. Класс `System.CSharp.RuntimeBinder.RuntimeBinderException`.
37. Перегрузка операторов в языке C#.
38. Принцип инверсии управления. Подход внедрения зависимостей.
39. Принцип инверсии управления. Контейнер внедрения зависимостей `DryIoc` / `Autofac` / `Windsor Castle` / `Unity` / `IServiceProvider` (на выбор). Регистрация зависимостей. Контексты регистрации (`Transient`, `Scoped`, `Singleton`). Подконтейнеры. Разрешение (`resolving`) зависимостей. Порождающие паттерны проектирования “Строитель”, “Одиночка”, “Абстрактная фабрика”.
40. Механизм отражений (`reflection`). Загрузка сборок в домен приложений. Реализация регистрации компонентов приложения на основе конфигурационного файла `appsettings.json`.

Часть 2. Windows Presentation Foundation (WPF).

41. Контейнеры компоновки `Grid`, `UniformGrid`.
42. Контейнеры компоновки `WrapPanel`, `StackPanel`, `DockPanel`.
43. Контейнер компоновки `Canvas`.
44. Контейнеры компоновки `TabControl`, `ItemsControl`.
45. Тэг `DataTemplate`.
46. Расширения разметки `x:Type`, `x:Static`.
47. Стандартные элементы управления: `TextBlock`, `Button`, `TextBox`, `Image` и их основные свойства.
48. Пользовательские элементы управления (`UserControl`).
49. Словари ресурсов. Слияние словарей ресурсов. Расширения разметки `StaticResource`, `DynamicResource`.
50. Стили. Наследование стилей. Установка значений свойств и обработчиков событий в стилях.
51. Свойство зависимости `Template`, тэг `ControlTemplate`.
52. Триггеры свойств зависимости `Trigger`, `MultiTrigger`.
53. Триггеры данных `DataTrigger`, `MultiDataTrigger`.
54. Триггеры событий `EventTrigger`.

55. Свойство зависимости. Метод Register класса System.Windows.DependencyProperty. Класс System.Windows.DependencyObject.
56. Присоединяемое свойство зависимости. Метод RegisterAttached класса System.Windows.DependencyProperty. Класс System.Windows.DependencyObject.
57. Метаданные свойств зависимости. Классы System.Windows.PropertyMetadata, System.Windows.UIPropertyMetadata, System.Windows.FrameworkPropertyMetadata, назначение.
58. Метаданные свойств зависимостей. Делегаты System.Windows.PropertyChangedCallback, System.Windows.ValidateValueCallback, System.Windows.CoerceValueCallback, назначение.
59. Привязки. Параметры привязок: Source, Path, Mode, TargetNullValue, FallbackValue, Converter, ConverterParameter.
60. Привязки. Расширение разметки RelativeSource. Параметры относительной привязки: Mode, AncestorType, AncestorLevel.
61. Интерфейс System.Collections.Specialized.INotifyCollectionChanged, назначение и предоставляемый функционал. Класс System.Collections.ObjectModel.ObservableCollection, назначение и примеры использования.
62. Команды. Интерфейс System.Windows.Input.ICommand, методы интерфейса. Поведенческий паттерн проектирования “Команда”.
63. Класс RelayCommand, его особенности, реализация и применение.
64. Конвертеры значений. Интерфейс System.Windows.Data.IValueConverter, назначение, примеры реализации и использования.
65. Конвертеры значений. Интерфейс System.Windows.Data.IMultiValueConverter, назначение, примеры реализации и использования.
66. Класс System.Windows.Markup.MarkupExtension, назначение, пример использования при наследовании (с реализацией).
67. Архитектурный паттерн Model-View-ViewModel. Интерфейс System.ComponentModel.INotifyPropertyChanged, предоставляемый интерфейсом функционал. Свойство зависимости DataContext.

Часть 3. Системы контроля версий.

68. Назначение систем контроля версий.
69. Понятие репозитория. Локальный репозиторий, удалённый репозиторий. Создание репозитория (init), клонирование репозитория (clone).
70. Система контроля версий git. Понятие ветки (branch). Создание и удаление локальных и удалённых веток. Переключение между ветками (checkout).

71. Система контроля версий git. Индексирование изменений (add). Фиксация изменений (commit). Зачем нужен commit message. Откат локально зафиксированных изменений (commit –amend).
72. Система контроля версий git. Отправка изменений в удалённый репозиторий (push). Что такое force push, rebase.
73. Система контроля версий git. Скачивание изменений из удалённого репозитория (fetch, pull, различия).
74. Система контроля версий git. Слияние веток (merge). Почему merge лучше делать во вливаемой ветке, а не в ветке в которую производится вливание. Конфликты при слиянии (merge conflict), автоматическое и ручное разрешение конфликтов.
75. Система контроля версий. Перенос коммита (cherry-pick), назначение, когда лучше использовать вместо merge.
76. Система контроля версий git. Прятание изменений (stash), назначение. Создание и применений спрятанных изменений.
77. Система контроля версий git. Тэги (tag), назначение.
78. Система контроля версий git. Назначение файла .gitignore.