1.Koriscenje tipova?

2.Konvertovanje vrednosti tipova?

3.Vrednosna varijabla, referentna varijabla, boxing, unboxing?

4.Sporedni efekti operatora?

5.Operator XOR, svrha koriscenje?

6.Class:

Svaka klasa moze da ima svoj cs?

1.Vehicle minivan; // declare a reference to an object

minivan = new Vehicle(); // allocate a Vehicle object

Konkretno?

2,Vehicle minivan = new Vehicle();

**minivan** does not, itself, contain the object. Instead, it contains a reference to the object.

3.Method osnove, modifikatori pristupa, void ne vraca vrednost.Po difoltu PRIVATE.

4.Method vs parametiziran method koja je razlika?

5.Method parametri?

6.Return a value from a method?

7.Constructors vs parameterized constructor?

8.What does each object have its own copy of?non static atributes.

9.Destructors zasto koristiti?

10.Array:

1.Zasto koristi nizove kad su Liste prilagodljivije?

2.Multimenzionalni cemu sluze?

3. Jagged arrays primena?

4.FailSoftArray ?

11.Gde se koriste Queue I stack?

12.Kako se citaju stringovi?

13.Ref and out?

16. Invoking an Overloaded Constructor Through this.Primena, objasnjenje?

17.Rekursija za racunske operacije?

18.Static metod primena,?

19.Static konstruktor I klase?

20. Overloading Binary Operators

22.Overloding Index and FailSoftArray?

23.Construcori I nasledjivanje?Kad se koji konstruktor poziva(base, *derived*)?

24. Virtual Methods and Overriding?

25.Polimorfizam?

26.Abstrakne klase I metode primena?

27.Object class razlicite funkcije klase?

28.Exception: Runtime?

29.The System.Exception Class

30.Exception-Handling:try, catch, finally.?

31.Using Multiple catch Clauses.

32.Try Blocks Can Be Nested, nema finally?

33.Multiple Catch blok je moguc?page 374/ 391

34.Throw exception?

Opening and Closing a File:

35.U primeru na 418, u try catch bloku, nema finnaily?

36.Pisanje file, sta znaci(423):

try {

do {

i = fin.ReadByte();

if(i != -1) fout.WriteByte((byte)i);

} while(i != -1);

} catch(IOException exc) {

Console.WriteLine(exc.Message);

37.Gde se nalazi ovaj txt file(page 428)”

try {

log\_out = new StreamWriter("logfile.txt");

38.Poredjenje fajlova(431 strana)?

39.as the core of C#.Zasto?, Da li je to dovoljno?

-Delegates and Events:

Namespaces:

Using.

40.Uz pomoc objekat generics I podatak su povezani direktno.Sta to znaci?

41. Generics Improve Type Safety, kako?

42. Constrained Types=ograniceni tipovi, smisao, funkcija?

43. The Reference Type and Value Type Constraints?

44.Multiple constrains, svrha?

45.TrafficLight:

public class TrafficLight : ITrafficCollor

{

public Lightcolour State

{

get;

private set;

}

public TrafficLight()

{

State = Lightcolour.Red;

}

public Lightcolour Next()

{

Lightcolour light;

for (light = Lightcolour.Red; light <= Lightcolour.Yellow; light++)

return light;

}

public Lightcolour SetBlinkingYellow()

{

Lightcolour.BlinkingYelow;

}

public Lightcolour SetInitialState(bool stop)

{

stop == true ? Lightcolour.Red : Lightcolour.Green;

return Lightcolour;

}

}

----------------------------------------------------------------------------

46.Attributes?

Attributes are C# language elements that decorate program elements with additional metadata that describes the program. This metadata is then evaluated at different places, such as runtime or design time for various purposes. The examples in this lesson showed how the*ObsoleteAttribute* attribute could be used to generate compile time warnings for deprecated code. Through applying the *DllImportAttribute*attribute, you could see how to apply both positional and named parameters to an attribute. Attributes may also be used to decorate various different types of program elements with a target descriptor. The example applied the *CLSCompliantAttribute* attribute to an entire assembly. However, it could have also been applied to different program elements with applicable target descriptors to limit its scope.

47. The Bitwise Operators?

48.Inkrement decrement razlika izmedju ++I I i++, kao I i— I –i.

49.Indexers smisao, overloading, korist=prednosti.

50.Exceptions:

// **Rethrow an exception.**

using System;

class Rethrow {

public static void GenException() {

// Here, numer is longer than denom.

int[] numer = { 4, 8, 16, 32, 64, 128, 256, 512 };

int[] denom = { 2, 0, 4, 4, 0, 8 };

for(int i=0; i < numer.Length; i++) {

try {

Console.WriteLine(numer[i] + " / " +

denom[i] + " is " +

numer[i]/denom[i]);

}

catch (DivideByZeroException) {

Console.WriteLine("Can't divide by Zero!");

}

catch (IndexOutOfRangeException) {

Console.WriteLine("No matching element found.");

throw; // rethrow the exception

}

}

}

}

Rethrow the exception.

374 C# 3.0: A Beginner’s Guide

class RethrowDemo {

static void Main() {

try {

Rethrow.GenException();

}

catch(IndexOutOfRangeException) {

// recatch exception

Console.WriteLine("Fatal error -- program terminated.");

}

}

}

51.instance variables =atribut?

52.Zasto?

// Create a query that groups websites by top-level domain name,

// but select only those groups that have more than two members.

// Here, ws is the range variable over the set of groups

// returned when the first half of the query is executed.

var webAddrs = from addr in websites

where addr.LastIndexOf(".") != -1

group addr by addr.Substring(addr.LastIndexOf("."))

into ws

where ws.Count() > 2

select ws;

53.

Lift:

Int?

54.Volatile??

55.

Objasnjenje koda:

private int FindClosestCalledFloor(int currFloor, bool[] buttons)

{

int ret = -1;

int minDistance = int.MaxValue;

for (int i = 0; i < buttons.Length; i++)

{

if (buttons[i])

{

int dist = Math.Abs(currFloor - i);

if (dist < minDistance)

{

minDistance = dist;

ret = i;

}

}

}

return ret;

}

56. The full error says "Unable to add data connection. Could not load file or assembly 'Microsoft.SqlServer.Management.Sdk.Sfc, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' or one of its dependencies. The system cannot find the file specified" as shown below

Could not load file or assembly Microsoft.SqlServer.management.sdk.sfc version 11.0.0.0

57.

SELECT

name,

CONCAT(

ROUND (

population / (select max(population) from world where continent = 'Europe')

\* 100, 0

) , '%'

) as population

from world

WHERE continent = 'Europe'

SELECT continent, name, population FROM world x

WHERE population >= ALL

(SELECT population FROM world y

WHERE y.continent=x.continent

AND population>0)

Web API

57.HTTP protocol?

58.Koraci kreiranja WEB API nisu isti?

59.Problem sa EntityFrameWorkom?

60.EntityFramework ogroman broj gresaka?

61.Anti virus izbrisao deo DLL-a.

61. api/CreateBetList ne prolazi.

62.Fidler tumacenje gresaka??

63. ServicesFactory

64.UserService sve preko DBEntities?

Kako se koriste storovane procedure u tom slucaju?

65.Sta tacni radi?

actual creator

66.Entity Framework:

namespace BetApplicationDLL

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Data.Entity.Core.Objects;

using System.Data.Entity.Core.Objects.DataClasses;

using System.Linq;

67.

Global:

GlobalConfiguration.Configure(WebApiConfig.Register);

68. WebAPIUserClient :

public class WebAPIUserClient : AddUserService

{

private static readonly string WEB\_API\_BASE\_URL = ConfigurationManager.AppSettings["SkipWebBaseAPI"];

public IEnumerable<spSearch\_Result> Search(string letter)

{

using (var client = new HttpClient())

{

client.BaseAddress = new Uri(WEB\_API\_BASE\_URL);

var responseTask = client.PostAsJsonAsync<string>("Search", letter);

responseTask.Wait();

var result = responseTask.Result;

if (result.IsSuccessStatusCode)

{

var readTask = result.Content.ReadAsAsync<IEnumerable<spSearch\_Result>>();

readTask.Wait();

var ret = readTask.Result;

return ret;

}

return null;

}

}

}

69.

SearchUser App.Config:

<appSettings>

<add key="SkipWebAPI" value="false"/>

<add key="WebAPIBaseURL" value="http://localhost: 51420/api/"/>

</appSettings>

70.

GlobalConfiguration.Configure(WebApiConfig.Register);

71.

WebApiConfig:

public static void Register(HttpConfiguration config)

{

// Web API routes

config.MapHttpAttributeRoutes();

config.Routes.MapHttpRoute(

name: "DefaultApi",

routeTemplate: "api/{controller}/{id}",

defaults: new { id = RouteParameter.Optional }

);

}

72.

IServicesFactory:

public class ServiceFactoryCreator

{

private static readonly bool \_skipWebApi;

static ServiceFactoryCreator()

{

string skip = ConfigurationManager.AppSettings["SkipWebAPI"];

if (!bool.TryParse(skip, out \_skipWebApi))

{

\_skipWebApi = false;

}

}

public static IServiceFactory CreateServicesFactory()

{

return \_skipWebApi ? (IServiceFactory)new InProcessServiceFactory() : (IServiceFactory)new WebAPIServiceFactory();

}

73. public class AddUserServiceController

[HttpPost]

[Route("api/AddUser")]

public bool AddUser(User user)

{

AddUserService us = new AddUserService();

bool state = us.AddUser(user);

return state;

}

[HttpPost]

[Route("api/AddAccount")]

public bool AddAccount(UserAccount useraccount)

{

AddUserService us = new AddUserService();

bool state = us.AddAccount(useraccount);

return state;

}

[HttpPost]

[Route("api/Search")]

public IEnumerable<spSearch\_Result> Search([FromBody]string letter)

{

AddUserService us = new AddUserService();

var state = us.Search(letter);

return state;

}

74.