В билете 3 вопроса: 2 теоретических и 1 практический.

# Теоретические вопросы экзамена по программированию.

1. Основные понятия программирования. Виды ЯП по назначению (системного,

Прикладного и т.п.)

1. Преимущества, недостатки платформы .NET.
2. Понятие общеязыковой среды исполнения (Common Language Runtime - CLR).
3. Управляемый (managed) и неуправляемый код.
4. Понятие Just In Time (JIT) компиляции.
5. Сравнительная характеристика языков C++ и C#.
6. Типы данных C#. Ссылочные типы и типы значений. Карты памяти для обоих видов.
7. Принципы объектно-ориентированного программирования (ООП).
8. Определение класса C#, принцип инкапсуляции. Области видимости объектов.
9. Конструкторы. Виды, поведение конструкторов, другие способы инициализации классов. Локальные и глобальные объекты.
10. Отношение наследования для классов. Понятие полиморфизма.
11. Класс Object. Перегрузка методов класса Object.
12. Абстрактные классы, интерфейсы. Назначение, отличия.
13. Универсальные (обобщенные) типы.
14. Хранение данных. Основные типы коллекций.
15. Язык запросов LINQ.
16. Делегаты. Использование делегатов при определении событий.
17. Делегаты. Использование лямбда – выражений.
18. Обработка ошибок. Типы ошибок.
19. Обработка ошибок. Конструкция try{ } catch { } finally.
20. Управление памятью. Сборка мусора в .Net.
21. Регулярные выражения.

Практические задания

1. Написать классы (без методов, только данные), реализующие следующие абстракции: Преподаватель, дисциплина

В классе описать требуемый набор полей данных (не менее 5). Переменные должны отражать сущность абстракции применительно к области образования/университетским системам учета.

1. Создайте класс “Рациональное число”. Число представляйте как соотношение P/Q, где P,Q – целые числа, Q ≠ 0, например 2/3, –6/7, 8/2, 10/1. Напишите функции для корректного сравнения объектов заданного типа
2. Создайте класс “Рациональное число”. Число представляйте как соотношение P/Q, где P,Q – целые числа, Q ≠ 0, например 2/3, –6/7, 8/2, 10/1. Перегрузите операторы сложения, вычитания объектов созданного класса.
3. class Teacher

{

string name; //имя, не может быть пустым >2 букв

string gender; //пол, одно из 2 значений: “m” или “f”

double experience; //стаж работы >= 0

DateTime birthDate; //дата рождения

List<string> disciplines; //список преподаваемых дисциплин

}

Создайте 2 конструктора класса – один по умолчанию, один с параметрами, в конструктор с параметрами включите проверку вводимого значения.

Покажите пример создания объектов класса

1. Определить тип возвращаемого значения для следующих функций и указать почему. Саму функцию писать не нужно.

* Функция, подсчитывающая возраст человека на основе текущей даты и даты рождения;
* Функция, сохраняющая в файле данные класса;
* Функция, выполняющая объединение двух строк (например, было 2 строки: “Иванов “ и “хороший человек”, станет “Иванов хороший человек”;
* Функция, проверяющая, является ли треугольник равносторонним;
* Метод, рассчитывающий значение площади окружности.

1. Укажите ошибки в следующем объявлении класса:

clas Triangle! //класс «треугольник»

{

int \_a, \_b, \_c; //стороны треугольника

public bool IsValid {

get() //стороны имеют допустимую длину?

{

int result = 0.5;

if( \_a + \_b <= \_c || b + c <= a || \_a + \_c <= \_b )

return 1;

return false;

}

}

int GetPerimeter() //расчет периметра

{

int result = \_a + \_b + \_c;

}

string GetName()

{ return “Треугольник” }

}

1. Реализовать интерфейс IEquation – определяющий поведение функции вида y = f(x) – функции одной переменной. На базе интерфейса реализовать 2 класса для уравнений вида a\*cos( b\*x+c ), а также a\*x + b
2. Для некоторого интерфейса IEquation, определяющего поведение функции f(x,y) написать класс, численно решающий дифференциальное уравнение вида , при заданных начальных значениях . Для решения можно использовать метод Эйлера, для которого .
3. Опишите интерфейс IEquation – определяющий поведение функции вида y = f(x) – функции одной переменной. На базе интерфейса - 2 класса для уравнений вида a\*x\*x + b\*x + c, а также sin(a\*x)/x
4. Создайте абстрактный класс *Геометрическая фигура*, предусмотрите в нем поля/свойства, общие для всех геометрических фигур, например координаты центра фигуры, функцию рисования. На базе этого класса создайте конкретный класс «прямоугольник». Переопределите методы рисования и ToString() для этого класса. Рисование фигуры можно проводить попиксельным закрашиванием площади, занятой фигурой: Bitmap image; image.SetPixel( координата x, координата y, цвет ).
5. Создайте базовый класс Геометрическая фигура, предусмотрите в нем общие для всех фигур поля/свойства, например координаты центра фигуры, цвет, предусмотрите возможность инициализации этих полей с помощью конструктора.

На базе этого класса создайте 2 новых на выбор: прямоугольник, трапеция, треугольник и окружность, либо другую геометрическую фигуру. В этих классах должны быть свои особые поля, характерные только для этого типа объектов, например радиус для окружности. Один из классов сделайте запечатанным (sealed).

1. Создайте универсальный класс Storage<T>, который будет представлять из себя хранилище на 10 элементов, основанное на массиве данных типа T. Определите требуемые свойства, методы работы с классом, в том числе функции добавления элемента и удаления последнего. Тип T должен быть классом.
2. Дан класс:

class University

{

public string Name { get;set; }

public int StudentsCount { get;set; }

public double Rating { get;set; }

public string City { get;set; }

public string [] Faculty { get;set; }

}

Создан список университетов:

List< University> univers = new List<University>(){ … };

Задача: выбрать названия университетов, находящихся в Москве, в которых обучается больше 500 студентов, рейтинг которых > 4.5

1. Дан список объектов типа Student. С помощью LINQ запроса выберите темы дипломов всех студентов факультета экономики, у которых средняя оценка > 4. Классы Student и Project описываются следующим образом:

class Student{

public string Name { get;private set;} //имя

public decimal Rating{ get;private set;} //средняя оценка

public string Faculty { get;} //факультет

public Project Diploma {get;} //диплом

}

class Project{ public string Theme{get;set;}}

1. Реализовать операторы сложения, вычитания для класса Деньги, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения.
2. Используя регулярное выражение найти стоимости продуктов в исходном тексте. Пример текста: "Добро пожаловать в наш магазин, вот наши цены: 1 кг. яблок - 20 руб. 2 кг. апельсинов - 30 руб. 0.5 кг. орехов - 50 руб.". Ожидаемый вывод:

1 кг. яблок - 20 руб.

2 кг. апельсинов - 30 руб.

0.5 кг. орехов - 50 руб.

1. Написать регулярное выражение, позволяющее найти фразу вида «студент группы ПИБ-XXX Иванов Иван Иванович». (отчество может отсутствовать).