**Михайловский Александр Станиславович:**

1. Какие есть типы данных в **C#** и как они хранятся в памяти?

Типы значений, а также ссылочные типы данных, которые включают в себя:  
целочисленные, вещественные,логические, символьные, ссылочные типы данных, перечесления и структуры. Ссылочные типы данных хранятся в куче, а типы значений в стеке.

1. Что такое явное/неявное преобразование типов? Приведите пример.  
   Неявное преобразование происходит автоматически, когда значение одного типа данных присваевается переменной другого типа, который может содержать более широкий диапазон значений.

Неявное преобразование:  
int n1 = 1;

double n2 = n1;

Явное преобразование производится вручную с использованием оператора приведения типов. В данном случае увеличивается риск ошибок, а также потери данных

Явное преобразование:

double n1 = 3.14;

int n2 = (int)n1;

1. Что такое операция упаковки/распаковки? Приведите пример этих операций.  
   Операция упаковки и распаковки связаны с преобраованием перемен ссылочного типа в значимые и обратно

boxing:

int n1 = 1;

object object1 = n1;

После упаковки переменная object1 имеет ссылку на объект.

unboxing:

object object1 = 1;

int n1 = (int)object1;

После распаковки переменная n1 имеет значение.

1. Чем отличается логический оператор **&** от **&&**? Приведите пример.  
   Оператор & выполняет побитовое логическое И и вычисляет оба операнда

bool result1 = (3<2) & (8>6);

Оператор && имеет схожий функционал, но если левый операнд принимает значение false, то правый операнд не вычисляется  
  
bool result1 = (3<2) && (5>6);

1. Чем отличается цикл **while** от **do-while**?  
     
   В цикле do-while сначала выполняется определённый код цикла в do, а потом происходит проверка условия в while

В цикле while сначала проверяется истинность условия и, если условие истинно - то выполняется код цикла.

1. Чем отличается **foreach** от **for**?  
     
   Цикл foreach используется для перебора каждого элемента заданного набора, а for выполняет операцию n раз
2. В чём отличие **break** от **continue**?  
     
   Оператор break используется тогда, когда нам необходимо выйти из цикла.

Оператор continue применяется тогда, когда мы хотим пропустить определённую итерацию.

1. Объявить двумерный массив одновременно с его инициализацией.  
     
   int[,] myarr = new int[,]

{  
{1,2,3},  
{4,5,6},  
{7,8,9},  
};

1. В чем отличие класса от объекта? Приведите пример класса и объекта.  
     
   Класс является описанием для создания объектов. Он определяет поведение(методы) и состояние(поля) объектов.

Например, мы имеем класс автомобиль с полями марка, модель, цвет и методы(запуск двигателя авто и т.п.)  
  
Объект представляет собой экземпляр класса, созданный по шаблону класса.  
Например, если мы имеем класс автомобиль, то любой определённый автомобиль будет объектом этого класса.

1. Приведите пример автосвойства и во что оно разворичивается?   
   public string Model { get; set;}

Данная запись может быть развёрнута как:  
  
public string Model  
{  
get  
{  
return Model;  
}  
set  
{  
Model = value;  
}  
}

1. Поддерживает ли **C#** множественное наследование? Приведите пример.  
   C# поддерживает только множественное наследование интерфейсов  
     
   interface Ibla  
   {

…  
}  
Intefrace Iblabla  
{  
…  
}  
class blablalba : Ibla, Iblabla  
{  
…  
}

1. Какое преимущество использования класса **StringBuilder** перед **String**?  
     
   Мы можем использовать StringBuilder тогда, когда предлопалается множество операций над текстом большого объёма.
2. Какие существуют модификаторы доступа в **С#**? Опишите их назначение.  
     
   - public, публичный, такой компонент доступен из любого места в коде.  
   - internal, компоненты класса доступны из любого места в коде в той же сборке.  
   - protected, компонент доступен в классе или производных класса.

-private, компонент доступен только в рамках своего класса или структуры.

1. Как можно передавать параметры методу в **С#**? Модификаторы параметров.  
     
   Параметры методу можно передавать по значению, ссылке или с использованием модификтором параметров.
2. Что такое **var**? Как оно работает?  
     
   Оператор var объявляет переменную, опционально присваивая ей значение.
3. Отличаются ли **Equals** и operator **==** ? Если да, в чем отличие?  
     
   Equals - это метод, а == - это оператор   
   Equal может быть переопределён в классе  
   Для типов значений Equals сравнивает значения полей, а == сравнивает значения самих переменных.
4. Перечислите основыне концепции **ООП**. Коротко опишите, что они означают.  
     
   1. Абстракция - отделение концепции от её экземпляра  
     
   2. Наследование - способность объекта или класса базироваться на другом объекте или классе.  
     
   3. Полиморфизм - реализация задач одной и той же идеи разными способами(Один интерфейс - множество реализаций).  
     
   4. Инкапсуляция - размещение одного объекта или класса внутри другого для разганичения дотупа к ним.
5. Отличие **abstract** метода от **virtual** метода.  
     
   Абстрактный метод - это метод, который должен быть реализован для создания конкретного класса.  
     
   Виртуальный метод - это метод, который может быть переопределён в производном классе.
6. Назовите методы класса System.Object и их назначение.  
     
   Метод Equals(Object) определяет, равен ли указанный объект текущему.  
     
   Метод ToString() возвращает строковое представление текущего объекта.  
     
   Метод ReferenceEquals сравнивает две ссылки.

Хеширование - преобразование входного массива данных произвольной длинны в выходную битовую строку фиксированной длинны.

1. Что такое абстрактный класс? Коротко опишите.  
     
   Абстрактный класс - это класс, в котором хотя бы один из элементов объявлен абстрактным. Используется ключевое слово abstract. Переопределение реализуется при помощи override. Абстрактный класс может применять определённый шаблон проектирования для классов, которые его расширяют.
2. Назовите отличия между интерфейсом и абстрактным классом?  
     
   Абстрактные классы могут содержать конструкторы, в то время как интрерфейс не может. Абстрактный класс может хранить данные в полях, интерфейс не может хранить данных. При использовании абстрактных классов не может быть множественного наследования, При использовании интерфейсов класс может наследоваться от нескольких интерфейсов. Абстрактный класс может переопределять только некоторые элементы, а интерфейс должен реализовывать все элементы. Абстрактный класс - наследование только для классов, интерфейс может быть реализован структурой.
3. Как запретить наследование от класса?  
     
   Наследование от класса можно запретить с помощью ключевого sealed.
4. Привидите пример ограничения для обобщенных типов, и укажите как оно работает.
5. Перечислите 3 типа делегатов и когда использовать каждый их них.  
     
   Делегат Action представляет собой метод, который не возвращает значение. Он используется, когда требуется передать метод без возвращаемого значения.  
     
   Делегат Func представляет собой метод, который возвращает значение. Он используется, когда требуется передать метод, возвращающий значение, например для вычислений.  
     
   Делегат Predicate представляет собой метод, который принимает значение и возвращает булевое значение. Он используется для проверки условий.
6. Укажите минимум 5 структур для хранения данных и укажите одним предлодением, что структура представляет (Пример: Массив - индексированная последовательность данных).  
     
   double - целочисленное число с плавающей запятой.  
     
   int - 32-битное знаковое целое число  
     
   bool - булевое значение true или false.  
     
   DataTime - представляет собой дату и время системы.  
     
   char - одиночный символ Юникода.
7. Перечислите принцип SOLID (если получится, дайте им краткое объяснение)  
     
   - Принцип единственной ответственности - для каждого класса должно быть определено единственное назначение, задача.

- Принцип открытости-закрытости. Програмные сущности должны быть открыты для расширения, но закрыты для изменений  
  
- Принцип подстановки Барбары Лисков. Методы, использующие некий тип, должны иметь возможность использовать его подтипы, не зная об этом.  
  
- Принцип разделения интерфейса. Предпочтительнее разделять интерфейсы на более мелкие тематические.  
  
- Принцип инверсии зависимостей. Система должна конструироваться на основе абстракций. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.

Задача 1:

Задан следующий трехмерный массив:

| 1  2  3  4  5 | int[,,] mas = { { { 1, 2 },{ 3, 4 } },                  { { 4, 5 }, { 6, 7 } },                  { { 7, 8 }, { 9, 10 } },                  { { 10, 11 }, { 12, 13 } }                }; |
| --- | --- |

С помощью циклов переберите все элементы этого массива и выведите их на консоль в следующем виде:

{{{1 , 2} , {3 , 4}} , {{4 , 5} , {6 , 7}} , {{7 , 8}, {9 , 10}} , {{10 , 11} , {12 , 13}}}

Задача 2:

Разработайте класс MyString для работы со строками символов. Определите для класса несколько конструкторов и public функцию реверса строки (входной парметр “abcdef” - результат “fedcba”, если кто-то в функции сделает вывод и не вернет результат, не засчитаю).

Задача 3:

Разработать метод расширения для типа DataTime возвращающий в результате своего выполнения строку в виде «День ‘DD’ месяца ‘MonthName’ года ‘YYYY+ 5501’ от сотворения мира.»

**Задача 4:** Разработайте класс MyStack, как стек целых чисел фиксированного размера, определите функции pop и push. При разработке нельзя использовать готовые классы , представляющие Stack, ArrayList, List<T> и аналогичные. В качестве внутреннего хранилища использовать только массивы.

Задача 5:

Разработайте базовый класс содержащий 2 поля и метод выводящий в консоль значения этих полей. Разработайте 2 производных класса каждый из которых должен содержать как минимум одно дополнительное поле и переопределять метод базового класса. Напишите код в методе Main() таким образом, чтобы изменяя только одну строку вы выбирали реализацию любого из производных классов

(

<строка кода для изменения>

<переменная>.<GetInfo()> // эта строка не изменяется, вообще никак, не тип и не имя

).

**Задача 6:**

Используя обобщения разработайте интерфейс для сохрания/чтения объекта произвольного типа в/из источник данных. Разработайте несколько классов реализующих интерфейс разработанный ранее. В качестве источника данных используйе коллекции.