|  |
| --- |
| №1  Введите номер(а) строки() без ошибок компиляции (номера строк указаны в комментариях).  using System;  public class Animal  {  }  public class Cat : Animal  {  }  public interface IStudent<out T>  {  }  public class Student<T> : IStudent<T>  {  }  class Program  {  static void Main(string[] args)  {  Animal animal = new Cat(); //1  IStudent<Animal> student = new Student<Animal>(); //2  IStudent<Cat> student2 = new Student<Animal>(); //3  IStudent<Animal> student3 = new Student<Cat>(); //4  }  }  Ответ: 124 |
| №2  В результате выполнения следующей программы:  using System;  namespace Application  {  public class Animal  {  }  public class Cat : Animal  {  }  public class Dog : Animal  {  }  public interface IStudent<out T>  {  void Move(T student);  }  public class Student<T> : IStudent<T>  {  public void Move(T student)  {  Console.Write("Run");  }  }  public class Program  {  static void Main(string[] args)  {  IStudent<Animal> student2 = new Student<Cat>();  student2.Move(new Dog());  Console.ReadKey();  }  }  }  Ответ: \*\*\*  на экран будет выведено:  *Примечание:*  *Если возникнет ошибка компиляции, выведите: \*\*\**  *Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---*  *Если возникнет ошибка исполнения или исключение, введите: +++* |

|  |
| --- |
| №3  В результате выполнения следующей программы:  using System;  public class Animal  {  public void Move()  {  Console.Write("Run!");  }  }  public class Zoo<T> where T : Animal  {  public void MoveAnimal(T animal)  {  animal.Move();  }  public T getNewAnimal()  {  T animal = new T();  return animal;  }  }  class Program  {  static void Main(string[] args)  {  Zoo<Animal> zoo = new Zoo<Animal>();  Animal dog = zoo.getNewAnimal();  zoo.MoveAnimal(dog);  Console.ReadLine();  }  }  на экран будет выведено:  *Примечание:*  *Если возникнет ошибка компиляции, выведите: \*\*\**  *Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---*  *Если возникнет ошибка исполнения или исключение, введите: +++*  Ответ: \*\*\* |
| №4  Введите номер(а) строки() без ошибок компиляции (номера строк указаны в комментариях).  public class Animal  {  }  public class Cat : Animal  {  }  public interface IStudent<T>  {  }  public class Student<T> : IStudent<T>  {  }  class Program  {  static void Main(string[] args)  {  Animal animal = new Cat();//1  IStudent<Animal> student = new Student<Animal>();//2  IStudent<Animal> student2 = new Student<Cat>();//3  }  }  Ответ: 12 |
| **№5**  Введите номер(а) строки() без ошибок компиляции (номера строк указаны в комментариях).  public class Animal  {  }  public class Cat : Animal  {  }  public interface IStudent<out T>  {  }  public class Student<T> : IStudent<T>  {  }  class Program  {  static void Main(string[] args)  {  Animal animal = new Cat();//1  IStudent<Animal> student = new Student<Animal>();//2  IStudent<Animal> student2 = new Student<Cat>();//3  }  }  Ответ: 123 |
| №6  В результате выполнения следующей программы:  using System;  using System.Collections;  class Program  {  static void Method()  {  ArrayList list = new ArrayList();  list.Add(1);  list.Add(3);  list.Add(101);  int sum = 0;  foreach (var item in list)  {  sum += item;  }  Console.WriteLine(sum);  }  static void Main(string[] args)  {  Method();  Console.ReadLine();  }  }  на экран будет выведено:  *Примечание:*  *Если возникнет ошибка компиляции, выведите: \*\*\**  *Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---*  *Если возникнет ошибка исполнения или исключение, введите: +++*  Ответ: \*\*\* |
| №7  Выберите верные утверждения   1. Обобщения автоматически обеспечивают типовую безопасность всех операций. 2. C# позволяет параметрам типов иметь типы по умолчанию. 3. Инвариантность - Отсутствие наследования между производными типами .   Ответ: 3 |
| №8  В результате выполнения следующей программы:  using System;  using System.Collections;  class Program  {  static void Method()  {  ArrayList list = new ArrayList();  list.Add(1);  list.Add(4);  list.Add(101);  list.Add("5");  int sum = 0;  foreach (var item in list)  {  sum += (int)item;  }  Console.WriteLine(sum);  }  static void Main(string[] args)  {  Method();  Console.ReadLine();  }  }  на экран будет выведено:  *Примечание:*  *Если возникнет ошибка компиляции, выведите: \*\*\**  *Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---*  *Если возникнет ошибка исполнения или исключение, введите: +++*  Ответ: +++ |
|  |