Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебный предмет
«Конструирование программ и языки программирования»

Инструкция
по выполнению лабораторной работы №14
«Разработка, отладка и испытание программ, с исследованием механизма наследования интерфейсов»

Минск 2024 г.

Лабораторная работа № 14

Тема работы: «Разработка, отладка и испытание программ, с исследованием механизма наследования интерфейсов»

1 Цель работы

Сформировать умения разрабатывать интерфейсы, и использовать их в программах, применять механизм множественного наследования интерфейсов.

2 Задание

Номер варианта соответствует номеру по спискув журнале.

Начиная с 17 учащегося номер варианта определяется по формуле: **N % 16 + 1**, где **N** – это ваш номер по списку, **%** - остаток от деления.

Создать и реализовать интерфейсы, также использовать наследование и полиморфизм для указанных предметных областей.

В классе должны быть реализованы:

- поля (закрытые или защищённые), нужное количество;
- свойства (для каждого поля);
- конструкторы (не менее двух);
- перегруженные операции;
- методы (нужное количество);
- метод ToString();
- статические поля, если они нужны;
- статические свойства (если есть статические поля).

1. interface Издание <- interface Книга <- class Справочник и Энциклопедия
2. interface Абитуриент <- interface Студент <- class Студент-Заочник.
3. interface Сотрудник <- interface Инженер <- class Руководитель.
4. interface Здание <- interface Общественное Здание <- class Театр.
5. interface Mobile <- interface Siemens Mobile <- class Model.
6. interface Корабль <- interface Военный Корабль <- class Авианосец.
7. interface Врач <- interface Хирург <- class Нейрохирург.
8. interface Корабль <- interface Грузовой Корабль <- class Танкер.
9. interface Мебель <- interface Шкаф <- class Книжный Шкаф.
10. interface Фильм <- interface Отечественный Фильм <- class Комедия.
11. interface Ткань <- interface Одежда <- class Костюм.
12. interface Техника <- interface Плеер <- class Видеоплеер.
13. interface Транспортное Средство <- interface Общественный Транспорт <- class Трамвай.
14. interface Устройство Печати <- interface Принтер <- class Лазерный Принтер.

15. interface Бумага <- interface Тетрадь <- class Тетрадь Для Рисования.

16. interface Источник Света <- interface Лампа <- class Настольная Лампа.

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Как только интерфейс будет определен, он может быть реализован в одном или не скольких классах. Для реализации интерфейса достаточно указать его имя после име ни класса, аналогично базовому классу. Ниже приведена общая форма реализации интерфейса в классе.

```
class имя_класса : имя_интерфейса {
   // тело класса
}
```

где имя_интерфейса — это конкретное имя реализуемого интерфейса. Если уж ин терфейс реализуется в классе, то это должно быть сделано полностью. В частности, реализовать интерфейс выборочно и только по частям нельзя.

В классе допускается реализовывать несколько интерфейсов. В этом случае все реа лизуемые в классе интерфейсы указываются списком через запятую. В классе можно наследовать базовый класс и в тоже время реализовать один или более интерфейс. В таком случае имя базового класса должно быть указано перед списком интерфейсов, разделяемых запятой.

Методы, реализующие интерфейс, должны быть объявлены как public. Дело в том, что в самом интерфейсе эти методы неявно подразумеваются как открытые, поэтому их реализация также должна быть открытой. Кроме того, возвращаемый тип и сигнатура реализуемого метода должны точно соответствовать возвращаемому типу и сигнатуре, указанным в определении интерфейса.

Ниже приведен пример программы, в которой реализуется интерфейс ISeries. В этой программе создается класс ByTwos, генерирующий последовательный ряд чисел, в котором каждое последующее число на два больше предыдущего.

```
// Реализовать интерфейс ISeries.
class ByTwos : ISeries {
   int start;
   int val;

   public ByTwos() {
      start = 0;
      val = 0;
   }

   public int GetNext() {
```

```
        val += 2;
        return val;
    }

    public void Reset() {
        val = start;
    }

    public void SetStart(int x) {
        start = x;
        val = start;
    }
}
```

Как видите, в классе ByTwos реализуются три метода, определяемых в интерфейсе ISeries. Как пояснялось выше, это приходится делать потому, что в классе нельзя реализовать интерфейс частично.

Ниже приведен код класса, в котором демонстрируется применение класса ByTwos, реализующего интерфейс ISeries.

```
// Продемонстрировать применение класса ByTwos, реализующего интерфейс.
using System;

class SeriesDemo {
    static void Main() {
        ByTwos ob = new ByTwos();
        for(int i=0; i < 5; i++)
            Console.WriteLine("Следующее число равно " + ob.GetNext());
        Console.WriteLine("\nСбросить");
        ob.Reset();
        for(int i=0; i < 5; i++)
            Console.WriteLine("Следующее число равно " + ob.GetNext());
        Console.WriteLine("\nНачать с числа 100");
        ob.SetStart(100);
        for(int i=0; i < 5; i++)
            Console.WriteLine("Следующее число равно " + ob.GetNext());
    }
}
```

Для того чтобы скомпилировать код класса SeriesDemo, необходимо включить в компиляцию файлы, содержащие интерфейс ISeries, а также классы ByTwos и SeriesDemo. Компилятор автоматически скомпилирует все три файла и сформирует из них окончательный исполняемый файл. Так, если эти файлы называются ISeries. cs, ByTwos.cs и SeriesDemo.cs, то программа будет скомпилирована в следующей командной строке:

```
>csc SeriesDemo.cs ISeries.cs ByTwos.cs
```

В интегрированной среде разработки Visual Studio для этой цели достаточно ввести все три упомянутых выше файла в конкретный проект C#. Кроме того, все три компилируемых элемента (интерфейс и оба класса) допускается включать в единый файл.

Ниже приведен результат выполнения скомпилированного кода.

```
Следующее число равно 2
Следующее число равно 4
Следующее число равно 6
Следующее число равно 8
Следующее число равно 10

Сбросить.
Следующее число равно 2
Следующее число равно 4
Следующее число равно 6
Следующее число равно 8
Следующее число равно 10
Начать с числа 100.
Следующее число равно 102
Следующее число равно 104
Следующее число равно 106
Следующее число равно 108
Следующее число равно 110
```

В классах, реализующих интерфейсы, разрешается и часто практикуется определять их собственные дополнительные члены. В качестве примера ниже приведен другой вариант класса ByTwos, в который добавлен метод GetPrevious(), возвращающий предыдущее значение.

```
// Реализовать интерфейс ISeries и добавить в
// класс ByTwos метод GetPrevious().
class ByTwos : ISeries {
   int start;
   int val;
   int prev;

   public ByTwos() {
      start = 0;
      val = 0;
      prev = -2;
   }

   public int GetNext() {
      prev = val;
      val += 2;
      return val;
   }
```

```
    public void Reset() {
      val = start;
      prev = start -2;
    }

    public void SetStart(int x) {
      start = x;
      val = start;
      prev = val - 2;
    }

    // Метод, не указанный в интерфейсе ISeries.
    public int GetPrevious() {
      return prev;
    }
  }
```

Как видите, для того чтобы добавить метод GetPrevious(), потребовалось внести изменения в реализацию методов, определяемых в интерфейсе ISeries. Но поскольку интерфейс для этих методов остается прежним, то такие изменения не вызывают никаких осложнений и не нарушают уже существующий код. В этом и заключается одно из преимуществ интерфейсов.

Как пояснялось выше, интерфейс может быть реализован в любом количестве классов. В качестве примера ниже приведен класс Primes, генерирующий ряд простых чисел. Обратите внимание на то, реализация интерфейса ISeries в этом классе ко ренным образом отличается от той, что предоставляется в классе ByTwos.

```
// Использовать интерфейс ISeries для реализации
// процесса генерирования простых чисел.
class Primes : ISeries {
  int start;
  int val;
  public Primes() {
  start = 2;
  val = 2;
  }
  public int GetNext() {
    int i, j;
    bool isprime;
    val++;
    for(i = val; i < 1000000; i++) {
      isprime = true;
      for(j = 2; j <= i/j; j++) {
        if((i%j)==0) {
```

```
                isprime = false;
                break;
            }
        }
        if(isprime) {
            val = i;
            break;
        }
    }
    return val;
}

public void Reset() {
    val = start;
}

public void SetStart(int x) {
    start = x;
    val = start;
}
}
```

Самое любопытное, что в обоих классах, ByTwos и Primes, реализуется один и тот же интерфейс, несмотря на то, что в них генерируются совершенно разные ряды чисел. Как пояснялось выше, в интерфейсе вообще отсутствует какая-либо реализация, поэтому он может быть свободно реализован в каждом классе так, как это требуется для самого класса.

5. Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм и теоретическую объектную модель решения задачи.
3. Запрограммировать полученные алгоритмы и объектную модель.

6. Форма отчета о работе

Лабораторная работа № ___

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Поясните, каким образом интерфейс может быть реализован в любом количестве классов?
2. Перечислите и охарактеризуйте стандартные интерфейсы.
3. Какой спецификатор доступа имеют по умолчанию все элементы интерфейса?

8. Рекомендуемая литература

1. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.

2. Прайс, М. Дж. C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.

3. Васильев, А.Н. Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.

4. Фримен, А. ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.