|  |  |  |
| --- | --- | --- |
| **Дата** | **Содержание выполняемой работы** | **Подпись**  **руководителя** |
| 03.04.2020 | **Практическая работа №12**  Тема: Обобщенные классы.  Наименование работы: Обобщенные классы, методы, делегаты, структуры, интерфейсы.  Цель: Сформировать навыки создания программ с использованием обобщенных методов. Сформировать навыки создания программ с использованием обобщенных классов. Сформировать навыки создания программ с использованием обобщенных интерфейсов.  Задание 1.  Изучил предложенную теорию. Кратко законспектировал.  Обобщения.  После появления первого выпуска платформы .NET программисты часто использовали пространство имен System.Collections для получения более гибкого способа управления данными в приложениях.  Обобщенные классы.  Ниже приведена общая форма объявления обобщенного класса:  class имя\_класса<список\_параметров\_типа> { // ...}  Ограниченные типы.  Допустим, что требуется создать метод, оперирующий содержимым потока, включая объекты типа FileStream или MemoryStream. На первый взгляд, такая ситуация идеально подходит для применения обобщений, но при этом нужно каким-то образом гарантировать, что в качестве аргументов типа будут использованы только типы потоков, но не int или любой другой тип. Кроме того, необходимо как-то уведомить компилятор о том, что методы, определяемые в классе потока, будут доступны для применения.  Иерархии обобщенных классов.  Обобщенные классы могут входить в иерархию классов аналогично необобщенным классам. Следовательно, обобщенный класс может действовать как базовый или производный класс.  Обобщенные методы.  В дополнение к обобщенным классам можно также определять обобщенные методы. В объявлении обобщенного метода присутствует обобщенный тип. Обобщенные методы могут быть определены внутри необобщенного класса.  Обобщенные структуры.  Подобно классам, структуры также могут быть обобщенными. Они очень похожи на обобщенные классы, за исключением возможности наследования. Итак, примером обобщенной структуры в .NET Framework является Nullable<T>.  Обобщенные делегаты.  Как и методы, делегаты также могут быть обобщенными. Ниже приведена общая форма объявления обобщенного делегата:  Delegate возвращаемый\_тип имя\_делегата<список\_параметров\_типа>(список\_аргументов);  Обобщенные интерфейсы.  Помимо обобщенных классов и методов, в C# допускаются обобщенные интерфейсы. Такие интерфейсы указываются аналогично обобщенным классам. Применяя обобщения, можно определять интерфейсы, объявляющие методы с обобщенными параметрами.  Задание 2.  Создал программы по образцу. Листинг кода переписал в дневник-отчет. Вставил скриншот выполнения программы.  Задание 2.1  Создание приложения с использованием обобщенного метода. Листинг кода переписал в Приложение 11. Результат выполнения программы представлен на Рисунке 24  jvjdkvdlf.png  Рисунок 24 – результат выполнения программы  Задание 2.2  Создание приложения с использованием обобщенного класса. Листинг кода переписал в Приложение 11. Результат выполнения программы представлен на Рисунке 25  dzjckaxa.png  Рисунок 25 – результат выполнения программы  Задание 2.3  Создание приложение с обобщенным интерфейсом. Листинг кода переписал в Приложение 11. Результат выполнения программы представлен на Рисунке 26  jckscns.png  Рисунок 26 – результат выполнения программы  **Контрольные вопросы**   1. Укажите, для чего предназначено обобщение.   Термин обобщение, по существу, означает параметризированный тип. Особая роль параметризированных типов состоит в том, что они позволяют создавать классы, структуры, интерфейсы, методы и делегаты, в которых обрабатываемые данные указываются в виде параметра.   1. Опишите обобщенные классы.   Ниже приведена общая форма объявления обобщенного класса:  class имя\_класса<список\_параметров\_типа> { // ...} А вот как выглядит синтаксис объявления ссылки на обобщенный класс:  имя\_класса<список\_аргументов\_типа>имя\_переменной  = new имя\_класса<список\_параметров\_типа> (спи  сок\_аргументов\_конструктора);   1. Опишите ограниченные типы.   Указывая параметр типа, можно наложить определенное ограничение на параметр. Это делается с помощью оператора where при указании параметра типа:  class имя\_класса<параметр\_типа>where параметр\_типа :  ограничения { // ...} где ограничения указываются списком через запятую.   1. Опишите обобщенные структуры.   Подобно классам, структуры также могут быть обобщенными. Они очень похожи на обобщенные классы, за исключением возможности наследования. Итак, примером обобщенной структуры в .NET Framework является Nullable<T>.   1. Опишите обобщенные методы.   В дополнение к обобщенным классам можно также определять обобщенные методы. В объявлении обобщенного метода присутствует обобщенный тип. Обобщенные методы могут быть определены внутри необобщенного класса.   1. Опишите обобщенные делегаты.   Как и методы, делегаты также могут быть обобщенными. Ниже приведена общая форма объявления обобщенного делегата:  delegate возвращаемый\_тип  имя\_делегата<список\_параметров\_типа>(список\_аргуме  нтов);   1. Опишите обобщенные интерфейсы.   Помимо обобщенных классов и методов, в C# допускаются обобщенные интерфейсы. Такие интерфейсы указываются аналогично обобщенным классам. Применяя обобщения, можно определять интерфейсы, объявляющие методы с обобщенными параметрами. |  |