

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

Учебный предмет
«Конструирование программ и языка программирования»

Инструкция
по выполнению лабораторной работы №23
«Разработка, отладка и испытание программ с применением сериализации»

Минск 2024 г.

Лабораторная работа № 23

Тема работы: «Разработка, отладка и испытание программ с применением сериализации»

1 Цель работы

Сформировать умение разрабатывать программы с применением сериализации объектов.

2 Задание

В задании по варианту лабораторной работы №19 организовать возможность загрузки данных из файла и сохранения с помощью механизма сериализации.

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Сериализация представляет процесс преобразования какого-либо объекта в поток байтов. После преобразования мы можем этот поток байтов или записать на диск или сохранить его временно в памяти.

При необходимости можно выполнить обратный процесс - десериализацию, то есть получить из потока байтов ранее сохраненный объект.

Чтобы объект определенного класса можно было сериализовать, надо этот класс пометить атрибутом Serializable:

```
[Serializable]
class Person
{
    public string Name { get; set; }
    public int Year { get; set; }

    public Person(string name, int year)
    {
        Name = name;
        Year = year;
    }
}
```

При отсутствии данного атрибута объект Person не сможет быть сериализован, и при попытке сериализации будет выброшено исключение `SerializationException`.

Сериализация применяется к свойствам и полям класса. Если нет необходимости, чтобы какое-то поле класса сериализовалось, то оно помечается атрибутом `NonSerialized`:

```
[Serializable]
class Person
```

```

{
    public string Name { get; set; }
    public int Year { get; set; }

    [NonSerialized]
    public string accNumber;

    public Person(string name, int year, string
acc)
    {
        Name = name;
        Year = year;
        accNumber = acc;
    }
}

```

В .NET можно использовать следующие форматы:

- бинарный
- SOAP
- xml
- JSON

Для каждого формата предусмотрен свой класс: для сериализации в бинарный формат - класс BinaryFormatter, для формата SOAP - класс SoapFormatter, для xml - XmlSerializer, для json - DataContractJsonSerializer.

Для классов BinaryFormatter и SoapFormatter сама функциональность сериализации определена в интерфейсе IFormatter.

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 понятие «Сериализация/десериализация»

2. Может ли объект, который подвергается десериализации, иметь конструктор без параметров?

3. Какие форматы можно использовать в .NET?

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.