

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра      Програмної інженерії

КУРСОВА РОБОТА  
ПОЯСНЮВАЛЬНА ЗАПИСКА  
з дисципліни “Об’єктно -орієнтоване програмування”  
Довідник любителя живопису

Керівник ,  
Студент гр. ПІ-15-2

Ляпота В.М.  
Фундукян А.А.

Комісія:	проф.	_____	Дудар З.В.
	асистент	_____	Ляпота В.М.
	лектор	_____	Бондарєв В.М.

Харків 2016

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**РАДІОЕЛЕКТРОНІКИ**

Кафедра: *Програмної інженерії*

Дисципліна: *Об'єктно-орієнтоване програмування*

Спеціальність: *Програмна інженерія*

Курс 1.

Група ПІ-15-2.

Семестр 2

**ЗАВДАННЯ**

*на курсовий проект студента*

***Фундукяна Артура Андрійовича***

1. Тема проекту: **«Справочник любителя живописи»**. Художники с анкетными данными и стилями. Картины со ссылкой на художников, датой создания, жанром. Коллекционеры и музеи: наличие оригиналов картин и копий. Аукционы и комиссионки: дата проведения, список выставленных шедевров и цены на них. Собственная коллекция.

2. Термін здачі студентом закінченого проекту: “28” - травня - 2016 р.

3. Вихідні дані до проекту: специфікація програми, методичні вказівки до виконання курсової роботи.

4. Зміст розрахунково-пояснювальної записки:

Вступ, специфікація програми, проектна специфікація, інструкція, висновки.

## КАЛЕНДАРНИЙ ПЛАН

	<i>Назва етапу</i>	<i>Термін виконання</i>
	Видача теми, узгодження і затвердження теми	01-03-2016 р.
	Формулювання вимог до програми	01-03-2016 – 12-03-2016 р.
	Розробка підсистеми зберігання інформації.	15-03-2016 – 24-03-2016 р.
	Розробка функцій зберігання та завантаження даних	15-03-2016 – 26-05-2016 р.
	Тестування і доопрацювання розробленої програмної системи.	20-04-2016 – 27-05-2016 р.
	Оформлення пояснювальної записки, додатків	20-05-2016 – 27-05-2016 р.
	Захист	28-05-2016 р.

Студент      Фундукян А.А.

Керівник      Ляпота В.М.

« 28 » травня 2016 р.

## РЕФЕРАТ

Пояснювальна записка до курсової роботи: 22 с., 6 рис., 1 додаток, 6 джерел.

Метою роботи є розробка програми “Справочник любителя живописи” на засадах об'єктно-орієнтованого програмування.

Методи розробки базуються на використанні середовища розробки Microsoft Visual Studio 2013, Windows Forms, платформи .NET Framework 4.5, мови програмування C#.

В результаті отримана програма під назвою “Справочник любителя живописи”, яка дозволяє зберігати перелік картин (їх назву, художника, дату створення, місце зберігання, жанр і автентичність), перелік художників (ім'я, роки життя та національність), музеї (назва, місцезнаходження та кількість картин з наведеного переліку). Є можливість редагувати всю підготовлену заздалегідь інформацію.

## ЗМІСТ

Вступ .....	6
1 Специфікація програми .....	7
1.1 Склад програми .....	7
1.2 Інтерфейс користувача .....	8
2 Інструкція по застосуванню .....	10
2.1 Інструкція користувача.....	11
ВИСНОВКИ.....	13
ПЕРЕЛІК ПОСИЛАНЬ.....	14
Додаток А – Код програми.....	28

## ВСТУП

З кожним днем сучасні технології стають все більш поширеними. Вони вже проникли в усі сфери людського життя. Настав час розробити свій проект для подальшого використання його в музеях, приватними колекціонерами, акціонерами і простими любителями живопису.

Оскільки даний проект є першим досить масштабним дітищем за весь час навчання, мета даної курсової роботи – вивчення основних принципів об'єктно-орієнтованого програмування, та оволодіння навичками розробки об'єктних програм. Зрозуміти окремі фази розробки програмного забезпечення і навчитися поєднувати їх в одне ціле – в свій проект.

Актуальність. У сучасних умовах життя, необхідно швидко реагувати на події. Оскільки наша майбутня професія - програміст, то отримання властивих навичок, поставлені тимчасові обмеження і масштаб роботи сильно розвивають навички і створюють умови схожі з тими, які будуть на майбутньому місці роботи. Також складений програмний продукт може послужити основою для більш великого проекту. Наданні програмою функції задовольняють основні потреби, які її потребують користувачів отже вже є рентабельним.

Метою курсового проектування є закріплення знань, набутих при вивченні дисципліни „Об'єктно-орієнтоване програмування”. Об'єктно - орієнтований підхід вимагає глибокого розуміння основних принципів, або, інакше, концепцій, на яких він базується. До числа основних понять ООП зазвичай відносять успадкування, інкапсуляцію і поліморфізм. Об'єктно - орієнтоване програмування в даний час є абсолютним лідером в області прикладного програмування.

Вивчення норм оформлення курсової роботи. Створення свого власного проекту і його захист. Розвиток ораторського досвіду.

# 1 СПЕЦИФІКАЦІЯ ПРОГРАМИ

Специфікацією даної програми є перегляд / редагування бази даних картин, художників, музеїв і аукціонів. Саме спрощення доступу до всієї важливої і потрібної інформації, для любителів живопису, колекціонерів і тд., є основною функцією і причиною розробки цієї програми.

## 1.1 Склад програми

Програма складається з набору класів і графічного інтерфейсу користувача.

Структура програми складається з 3 основних частин, які тісно взаємодіють між собою.

Одним з найважливіших методів програми є методи серіалізації і десеріалізації об'єктів. Оскільки дані методи містяться в декількох класах, так як різні об'єкти серіалізуються в різні файли, розглянемо їх на прикладі одного класу (рис. 1.1, рис. 1.2).

```
public void SaveDataContext()  
{  
    BinaryFormatter binaryFormatter = new BinaryFormatter();  
    using (FileStream fs = new FileStream(Path, FileMode.Create))  
    {  
        binaryFormatter.Serialize(fs, this);  
    }  
}
```

Рисунок 1.1 – Серіалізація

```
private void LoadDataContext()  
{  
    BinaryFormatter binaryFormatter = new BinaryFormatter();  
    using (FileStream fs = new FileStream(Path, FileMode.Open))  
    {  
        var dataContext = (DataContext)binaryFormatter.Deserialize(fs);  
        _instance = dataContext;  
    }  
}
```

Рисунок 1.2 – Десеріалізація

## 1.2 Інтерфейс користувача

Використовуються 4 основні вкладки і меню для взаємодії з системою:

1. «Картини» - надає користувачеві повний перелік картин з готової бази даних, дату їх створення, творця, який створив шедевр. Так само доступні жанр в якому написана картина, її справжність і справжнє місце знаходження (в якій галереї, музеї або колекції вона складається на сьогоднішній день).
2. «Художники» - список художників, дані про їх тривалості життя і національності.
3. «Музей» - зберігають в собі перелік музеїв, їх місце положення і кількість картин, які були згадані в довіднику.
4. «Аукціони» - володіють переліком поточних, або минулих аукціонів, і даними о картинах, які приймають участь в аукціоні.

Название	Создатель	Подлинность	Жанр	Дата создания	Местонахождение
Starry Night Over the Rhone	Vincent van Gogh	<input checked="" type="checkbox"/>	Landscape	01.01.1888	Musée d'Orsay, Paris
The Starry Night	Vincent van Gogh	<input checked="" type="checkbox"/>	Landscape	01.06.1889	Museum of Modern Art, New...
Olive Trees with the Alpilles i...	Vincent van Gogh	<input checked="" type="checkbox"/>	Still_Life	01.05.1889	Museum of Modern Art, New...
Wheatfield with Crows	Vincent van Gogh	<input checked="" type="checkbox"/>	Landscape	01.07.1890	Van Gogh Museum, Amsterd...
Sunflowers	Vincent van Gogh	<input checked="" type="checkbox"/>	Still_Life	01.06.1888	Private collection
The Persistence of Memory	Salvador Dalí	<input checked="" type="checkbox"/>	Impression	01.01.1931	Museum of Modern Art, New...
The Elephants	Salvador Dalí	<input checked="" type="checkbox"/>	Impression	01.01.1948	Private collection
Dream Caused by the Flight ...	Salvador Dalí	<input checked="" type="checkbox"/>	Impression	01.04.1944	Thyssen-Bornemisza Museu...
Portrait of a Young Woman ...	Peter Paul Rubens	<input checked="" type="checkbox"/>	Portrait	01.07.1609	Thyssen-Bornemisza Museu...
Venus at the Mirror	Peter Paul Rubens	<input checked="" type="checkbox"/>	Portrait	01.01.1615	Private collection
Infanta Isabella Clara Eugenia	Peter Paul Rubens	<input checked="" type="checkbox"/>	Portrait	01.01.1625	Kunsthistorisches Museum, ...
Virgin in Adoration before th...	Peter Paul Rubens	<input checked="" type="checkbox"/>	Portrait	01.01.0001 0:00	Private collection
Miracle of Saint Hubert	Peter Paul Rubens	<input checked="" type="checkbox"/>	Landscape	01.07.1890	Private collection
View at Rouelles	Oscar-Claude Monet	<input checked="" type="checkbox"/>	Landscape	26.05.2016 22:32	Private collection
In the Garden	Oscar-Claude Monet	<input checked="" type="checkbox"/>	Landscape	26.05.2016 22:32	Collection E. G. Buehrle, Zü...
Weeping Willow	Oscar-Claude Monet	<input checked="" type="checkbox"/>	Impression	26.05.2016 22:32	Kimball Art Museum, Fort W...
Rouen Cathedral at sunset	Oscar-Claude Monet	<input checked="" type="checkbox"/>	Impression	26.05.2016 22:32	Kimball Art Museum, Fort W...

Рисунок 1.2.1 – Головне вікно програми

Дані в таблицях можуть бути відредаговані шляхом зміни тексту за допомогою миші і клавіатури.



Також присутній інтерфейс для управління самим файлом. Кнопка в верхньому лівому кутку «Файл», натискання на яку відкриває цілий список функцій.

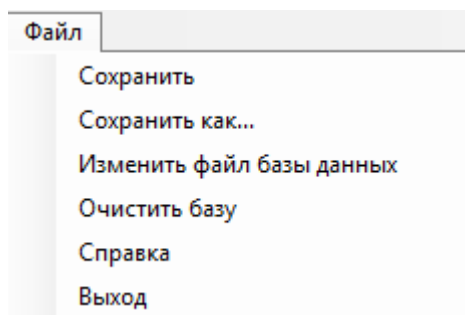


Рисунок 1.2.2 – Файл

1. Клавиша «Сохранить» - здійснює перезапис даних у файлі бази даних.
2. Клавиша «Сохранить как» - здійснює збереження файлу бази даних в нестандартній директорії для подальшого його використання
3. Клавиша «Изменить файл базы данных» - змінює шлях до довантажувати, стандартно, бази даних на новий.
4. Клавиша «Очистить базу» - повертає всі зміни до «default» стану.
5. Клавиша «Справка» - показує коротку інформацію про творця програми і про тему курсової роботи.
6. Клавиша «Выход» - здійснює автоматичне збереження внесених змін і завершення роботи з програмою.

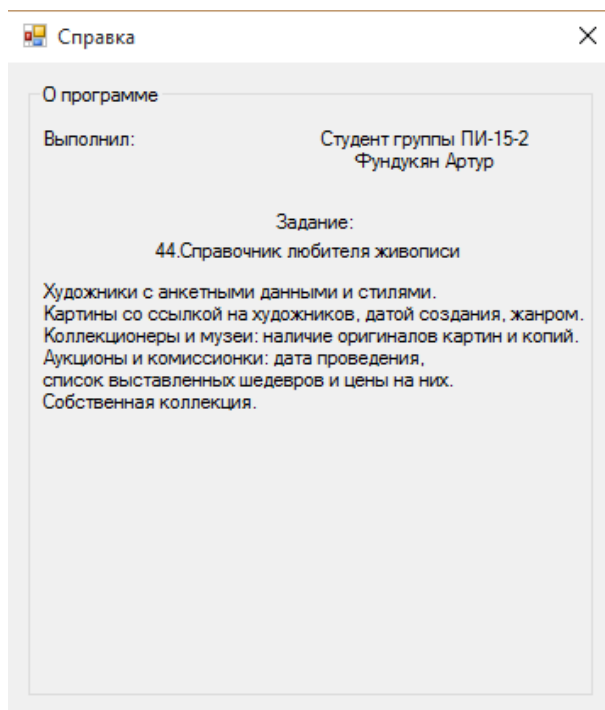


Рисунок 1.2.3 – Справка

## 2 ІНСТРУКЦІЯ ПО ЗАСТОСУВАННЮ

В даному розділі знаходиться інформація про особливості програми і її використання. Програма демонструє роботу з даними. Програма може бути використана в різних приватними колекціонерами, музеями, художніми галереями, аукціонами і т.д.

Після запуску програми відображується головне вікно (рис. 3.1)

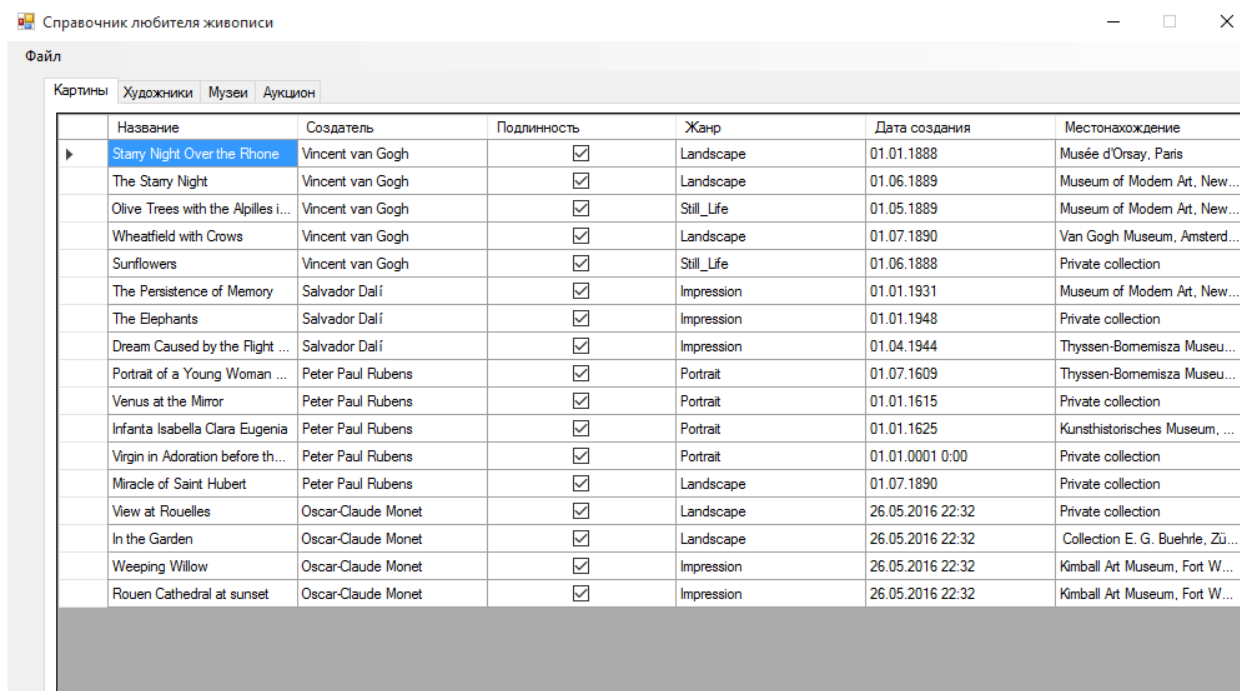


Рисунок 2.1 – Головне вікно програми

Саме воно є доступом користувача до усіх закладених до програми функцій.

## 2.1 Інструкція користувача

Використовуються 4 основні вкладки і меню для взаємодії з системою:

Після відкриття головного вікна програми (Рис. 3.1), у користувача відкривається широкий спектр можливості по отримання важливої інформації.

Натискання на вкладку «Картини» відкриває весь список картин, які були закладені в базу даних творцем даного програмного додатка. В даній таблиці, є можливість виправити / переробити надану інформацію, аж до зміни статусу картини з «Оригінал» на «Репродукція».

Наступна вкладка «Художники» призначена для тих, хто хоче знайти коротку інформацію про творця однієї з картин, представлених на «сторінках» даного довідника минулого вкладці.

Інформацію про музеї, які фігурують в одному зі списків можна отримати з вкладки «Музеї». Дані музеї є одними з найвідоміших у світі.

Остання вкладка, що надає інформацію - «Аукціони». Тут зберігається інформація про майбутні і минулі аукціони, про картини, які на них виставлені і вартість конкретного лоту.

Дані в таблицях можуть бути відредаговані шляхом зміни тексту за допомогою миші і клавіатури.

Присутній інтерфейс для управління файлом бази даних.

Для того, що б зробити якесь дію з файлом бази даних, необхідно натиснути на кнопку «Файл».

У ньому знаходяться кілька розділів.

Для збереження внесених змін необхідно натиснути клавішу «Сохранить».

Натискання на «Сохранить как» викличе діалогове вікно в якому користувач буде повинен вибрати місце положення для збереження файлу бази даних. Ця функція може бути корисна для тих, хто використовує кілька різних файлів з даними.

«Изменить файл базы данных» - проводить зміну файлу, який завантажується як основа, тобто, виправдовує функція показану в попередньому пункті.

Для повернення до попереднього стану файлу бази даних (default) існує кнопка «Очистить базу».

Безпечним варіантом виходу, творець програми вважає клавішу «Выход», натискання на яку автоматично робить перезапис кореневого файлу бази даних і відповідно завершає програму.

Даний функціонал є не повним і буде доопрацьовуватися в наступних ітераціях.

## ВИСНОВКИ

В результаті даної курсової роботи була розроблена програма "Довідник любителя живопису", призначена для перегляду картин, музеїв, аукціонів, художників і колекціонерів. Були закріплені знання, отримані за весь рік навчання на програмній інженерії. Проведена робота з налагодження і перевірки створеного продукту.

Проведена спроба захисту проекту. Отримано досвід створення і захисту масштабної роботи на приклад курсової.

Для вдосконалення готового програмного продукту є кілька напрямків. Перше - додавання нових функцій. Даний довідник можна розширити за допомогою функцій "додати" і "видалити" в інтерфейсі користувача.

Друге - удосконалення самого Інтерфейсу користувача. Windows Forms не є ідеальним рішенням для масштабної програми. Третє - вдосконалення програмної логіки

## ПЕРЕЛІК ПОСИЛАНЬ

1. 1. Бондарев, В.М. Объектно-ориентированное программирование на С#. Учебное пособие [Текст] / В. М. Бондарев. – Харьков : СМИТ, 2009. – 224 с.
2. Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#; Питер, 2007. - 656 с.
3. Робинсон, С.; Корнес, О.; Глинн, Д. и др. C# для профессионалов; М.: Лори, 2005. - 396 с
4. Шилдт, Герберт C# 4.0: Полное руководство; ; М.: Лори, 2011. - 1056 с
5. Уроки программирования с нуля. Си-шарп для чайников [Электронный ресурс]; URL: <http://mycsharp.ru/>
6. Microsoft Software Developer Network [Электронный ресурс]; URL: <https://msdn.microsoft.com/ru-ru/default.aspx>

## Додаток А

### Код програми

#### Клас «Program.cs»

```
static class Program
{
    /// <summary>
    /// Главная точка входа для приложения.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);

        DataContext.Instance.Initialize();

        Application.Run(new MainForm());
    }
}
```

#### Клас «Artist.cs»

```
[Serializable]
public class Artist
{
    public Artist()
    {
        Pictures = new List<Picture>();
    }
    public Artist(string name)
    {
        Name = name;
    }

    [DisplayName("Имя")]
    public String Name { get; set; }

    [DisplayName("Годы жизни")]
    public String LifeTime { get; set; }
    [DisplayName("Национальность")]
    public String Nationality { get; set; }

    public List<Picture> Pictures { get; set; }
    public List<ArtGenre> ArtGenres { get { return Pictures.Select(x =>
x.Genre).Distinct().ToList(); } }
}
```

## Клас «Auction.cs»

```
[Serializable]
public class Auction
{
    public Auction()
    {
        Items = new List<AuctionItem>();
    }

    public string Title { get; set; }
    public DateTime Date { get; set; }
    public List<AuctionItem> Items { get; set; }
}
```

## Клас «AuctionItem.cs»

```
[Serializable]
public class AuctionItem
{
    [Browsable(false)]
    public Picture Art { get; set; }

    [DisplayName("Название картины")]
    public string ArtTitle { get { return Art.Title; } }

    private int _price;

    [DisplayName("Цена")]
    public int Price
    {
        get { return _price; }
        set
        {
            if (value >= 0)
                _price = value;
        }
    }

    [DisplayName("Владелец")]
    public string Owner
    {
        get { return Art.Title; }
    }

    public AuctionItem() { }

    public AuctionItem(Picture item, int price)
    {
        Art = item;
        Price = price;
    }
}
```



## Клас «Collector.cs»

```
[Serializable]
public class Collector
{
    public Collector()
    {
        Collection = new List<Picture>();
    }
    public Collector(string name)
    {
        Name = name;
    }
    public string Name { get; set; }
    public List<Picture> Collection { get; set; }
}
```

## Клас «Museum.cs»

```
[Serializable]
public class Museum
{
    public Museum()
    {
        Pictures = new List<Picture>();
    }
    [DisplayName("Название")]
    public string Title { get; set; }
    [DisplayName("Местонахождение")]
    public string Place { get; set; }
    [DisplayName("Картины")]
    public List<Picture> Pictures { get; set; }
    [DisplayName("Кол-во картин")]
    public int PicturesCount { get { return Pictures.Count; } }
}
```

## Клас «Picture.cs»

```
[Serializable]
public enum ArtGenre
{
    History,
    Portrait,
    Genre,
    Landscape,
    Still_Life,
    Mythologic,
    Impression
}

[Serializable]
public class Picture
{
    public Picture()
    {
        Verified = true;
        Date = DateTime.Now;
    }

    [DisplayName("Название")]
    public string Title { get; set; }

    [DisplayName("Создатель")]
    public string Artist { get; set; }

    [DisplayName("Подлинность")]
    public bool Verified { get; set; }

    [DisplayName("Жанр")]
    public ArtGenre Genre { get; set; }

    [DisplayName("Дата создания")]
    public DateTime Date { get; set; }

    [DisplayName("Местонахождение")]
    public string Place { get; set; }
}
```

## Класс «DataContext.cs»

```
[Serializable]
public class DataContext
{
    private static DataContext _instance;

    public static DataContext Instance
    {
        get
        {
            if (_instance == null)
                _instance = new DataContext();
            return _instance;
        }
    }

    public List<Picture> Pictures { get; set; }
    public List<Artist> Artists { get; set; }
    public List<Auction> Auctions { get; set; }
    public List<Museum> Museums { get; set; }
    public List<Collector> Collectors { get; set; }
    public string Path { get; set; }

    private DataContext()
    {
        Path = System.IO.Path.Combine(Environment.CurrentDirectory + @"\db.dat");
    }

    public void Initialize()
    {
        Pictures = new List<Picture>();
        Artists = new List<Artist>();
        Auctions = new List<Auction>();
        Museums = new List<Museum>();
        Collectors = new List<Collector>();

        if (!CheckIfDataExists())
            InitializeContext();
        else
            LoadDataContext();
    }

    private void LoadDataContext()
    {
        BinaryFormatter binaryFormatter = new BinaryFormatter();
        using (FileStream fs = new FileStream(Path, FileMode.Open))
        {
            var dataContext = (DataContext)binaryFormatter.Deserialize(fs);
            _instance = dataContext;
        }
    }

    public void ForceClearDatabase()
    {
        if (File.Exists(Path))
            File.Delete(Path);

        InitializeContext();
    }
}
```

```

private void InitializeContext()
{
    Pictures = new List<Picture>()
    {
        new Picture() {Title = "Starry Night Over the Rhone", Genre =
ArtGenre.Landscape,Verified = true, Date = new DateTime(1888,01,01) , Artist = "Vincent
van Gogh", Place = "Musée d'Orsay, Paris"},
        new Picture() {Title = "The Starry Night", Genre =
ArtGenre.Landscape,Verified = true, Date = new DateTime(1889,06,01), Artist= "Vincent
van Gogh", Place = "Museum of Modern Art, New York" },
        new Picture() {Title = "Olive Trees with the Alpilles in the Background",
Genre = ArtGenre.Still_Life, Verified = true, Date = new DateTime(1889,05,01), Artist =
"Vincent van Gogh", Place = "Museum of Modern Art, New York" },
        new Picture() {Title = "Wheatfield with Crows", Genre =
ArtGenre.Landscape,Verified = true,Date = new DateTime(1890,07,01), Artist= "Vincent van
Gogh", Place = "Van Gogh Museum, Amsterdam" },
        new Picture() {Title = "Sunflowers", Genre = ArtGenre.Still_Life,Verified
= true,Date = new DateTime(1888,06,01), Artist= "Vincent van Gogh", Place = "Private
collection" },

        new Picture() {Title = "The Persistence of Memory", Genre =
ArtGenre.Impression, Verified = true,Date = new DateTime(1931,01,01), Artist = "Salvador
Dalí", Place = "Museum of Modern Art, New York" },
        new Picture() {Title = "The Elephants", Genre = ArtGenre.Impression,
Verified = true,Date = new DateTime(1948,01,01), Artist = "Salvador Dalí", Place =
"Private collection" },
        new Picture() {Title = "Dream Caused by the Flight of a Bee Around a
Pomegranate a Second Before Awakening", Genre = ArtGenre.Impression, Verified = true,Date
= new DateTime(1944 ,04,01), Artist = "Salvador Dalí", Place = "Thyssen-Bornemisza
Museum, Madrid" },

        new Picture() {Title = "Portrait of a Young Woman with a Rosary", Genre =
ArtGenre.Portrait, Date = new DateTime(1609,07,01), Artist = "Peter Paul Rubens", Place
= "Thyssen-Bornemisza Museum, Madrid" },
        new Picture() {Title = "Venus at the Mirror", Genre = ArtGenre.Portrait,
Date = new DateTime(1615,01,01) ,Artist = "Peter Paul Rubens", Place = "Private
collection" },
        new Picture() {Title = "Infanta Isabella Clara Eugenia", Genre =
ArtGenre.Portrait, Date = new DateTime(1625,01,01), Artist = "Peter Paul Rubens", Place
= "Kunsthistorisches Museum, Vienna." },
        new Picture() {Title = "Virgin in Adoration before the Christ Child",
Genre = ArtGenre.Portrait, Date = new DateTime(1615), Artist = "Peter Paul Rubens", Place
= "Private collection" },
        new Picture() {Title = "Miracle of Saint Hubert", Genre =
ArtGenre.Landscape, Date = new DateTime(1890,07,01), Artist = "Peter Paul Rubens", Place
= "Private collection" },

        new Picture() {Title = "View at Rouelles", Genre = ArtGenre.Landscape,
Artist = "Oscar-Claude Monet", Place = "Private collection" },
        new Picture() {Title = "In the Garden", Genre = ArtGenre.Landscape,
Artist = "Oscar-Claude Monet", Place = " Collection E. G. Buehrle, Zürich" },
        new Picture() {Title = "Weeping Willow", Genre = ArtGenre.Impression,
Artist = "Oscar-Claude Monet", Place = "Kimball Art Museum, Fort Worth, " },
        new Picture() {Title = "Rouen Cathedral at sunset", Genre =
ArtGenre.Impression, Artist = "Oscar-Claude Monet", Place = "Kimball Art Museum, Fort
Worth, " },

    };

    Artists = new List<Artist>()
    {

```

```

        new Artist() { Name = "Vincent van Gogh", Nationality = "Dutch", LifeTime
= "30 March 1853 - 29 July 1890", Pictures =
{Pictures[0], Pictures[1], Pictures[2], Pictures[4], Pictures[5] }},
        new Artist() { Name = "Salvador Dalí", LifeTime = "11 May 1904 - 23
January 1989 ", Nationality = "Spanish", Pictures = {Pictures[5], Pictures[6], Pictures[7]
}},
        new Artist() { Name = "Peter Paul Rubens", LifeTime = "28 June 1577 - 30
May 1640", Nationality = "Flemish", Pictures = {Pictures[8], Pictures[9],
Pictures[10], Pictures[11] }},
        new Artist() { Name = "Oscar-Claude Monet", LifeTime = "14 November 1840 -
5 December 1926 ", Nationality = "French", Pictures = {Pictures[5], Pictures[6],
Pictures[7] }},
    };

    var auctionItem = new AuctionItem() { Art = Pictures[0], Price = 500 };
    var auctionItem2 = new AuctionItem() { Art = Pictures[5], Price = 700 };
    var auctionItem3 = new AuctionItem() { Art = Pictures[3], Price = 900 };
    var auctionItem4 = new AuctionItem() { Art = Pictures[12], Price = 2700 };
    var auctionItem5 = new AuctionItem() { Art = Pictures[9], Price = 150 };
    Auctions = new List<AuctionItem>()
    {
        new Auction() {Title = "Under the tree", Date = new DateTime(2016, 5, 25),
Items = { auctionItem, auctionItem2, auctionItem5 } },
        new Auction() {Title = "Chestnut tree", Date = new DateTime(1997, 8, 15),
Items = { auctionItem3, auctionItem4 }
    }
    };

    Museums = new List<Museum>()
    {
        new Museum() {Title = "Museum of Modern Art", Place = "New York",
Pictures = { Pictures[1], Pictures[2], Pictures[4] } },
        new Museum() {Title = "Thyssen-Bornemisza Museum", Place = "Madrid",
Pictures = { Pictures[7], Pictures[8] } },
    };

    Collectors = new List<Collector>()
    {
        new Collector() {Name = "First collector", Collection =
Enumerable.Empty<Picture>().ToList()}
    };

    SaveDataContext();
}

public void SaveDataContext()
{
    BinaryFormatter binaryFormatter = new BinaryFormatter();
    using (FileStream fs = new FileStream(Path, FileMode.Create))
    {
        binaryFormatter.Serialize(fs, this);
    }
}

private bool CheckIfDataExists()
{
    if (File.Exists(Path))
    {
        return true;
    }
    return false;
}

```

```
public Artist GetArtistByName(string name)
{
    return Artists.FirstOrDefault(x => x.Name.Contains(name));
}
```