

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра вычислительной техники**

**ОТЧЕТ**  
**по лабораторной работе №8**  
**по дисциплине «Объектно-ориентированное программирование»**  
**ТЕМА: «Модульное тестирование приложения»**

Студент гр. 3314

\_\_\_\_\_

Кокорев С.С.

Преподаватель

\_\_\_\_\_

Гречухин М.Н.

Санкт-Петербург

2024

### **Цель работы.**

Знакомство с технологией модульного тестирования C# приложений с использованием библиотеки xUnit.

### **Задание на лабораторную работу.**

1. Проанализируйте классы приложения и определите, какие методы необходимо протестировать.
2. Напишите Unit-тесты для выбранных методов.
3. Запустите тесты и снимите с экрана скриншоты, иллюстрирующие выполнение тестов.

### **Задание курсовой работы.**

Задание 1. Разработать ПК для работников библиотеки. В ПК должны храниться сведения об имеющихся в библиотеке книгах и о читателях библиотеки. Библиотекарь могут потребоваться следующие сведения:

- какие книги закреплены за читателем;
- кто автор и как называется книга с заданным шифром.

Библиотекарь может вносить следующие изменения:

- запись нового читателя в библиотеку;
- пополнение библиотеки;
- списывание старой книги;
- изменение шифра книги.

Необходимо предусмотреть возможность выдачи справки о количестве читателей библиотеки и работе библиотеки за месяц (количество выданных книг, число записавшихся читателей).

## Перечень тестируемых методов

### 1) Метод получения строки для подключения к БД

```
public static string GetConnectionString(string filePath)
{
    var doc = XDocument.Load(filePath);

    string host = doc.Root.Element("Host").Value;
    string port = doc.Root.Element("Port").Value;
    string database = doc.Root.Element("Database").Value;
    string username = doc.Root.Element("Username").Value;
    string password = doc.Root.Element("Password").Value;

    string connectionString =
$"Host={host};Port={port};Database={database};Username={username};Password={password}";

    return connectionString;
}
```

### 2) Создание класса подключения к БД

```
private string DBConnectionString;
public DbSet<Visitor> Visitors { get; set; }
public DbSet<Book> Books { get; set; }
public DbSet<Author> Authors { get; set; }
public DbSet<LibraryTransaction> Transactions { get; set; }
public LibraryDbContext(string connectionString)
{
    DBConnectionString = connectionString;
}
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseNpgsql(DBConnectionString);
}
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    CreateAuthors(modelBuilder);
    CreateBooks(modelBuilder);
    CreateVisitors(modelBuilder);
    CreateTransactions(modelBuilder);
}
```

### 3) Метод добавления книги в БД

```
public void AddBookTest()
{
    var manager = new Manager();

    int oldBooksCount = manager.Books.Count;

    var newBook = new Book()
    {
        AuthorId = 1,
        BookAuthor = manager.Authors[0],
        Name = "TestBook",
        Year = 1010
    };
    manager.AddBook(newBook);

    int newBooksCount = manager.Books.Count;
```

```
    Assert.Equal(oldBooksCount + 1, newBooksCount);  
}
```

## **Код тестов**

### **1) Тест получения строки для подключения к БД**

```
public void GetConnectionString()  
{  
    var infoPath = "D:/proj/Visual Studio/LibManage/Other/DBConnectionInfo.xml";  
  
    var dbConnectionString = ConnectionStringGetter.GetConnectionString(infoPath);  
  
    Assert.NotNull(dbConnectionString);  
    Assert.NotEmpty(dbConnectionString);  
}
```

### **2) Тест создания класса подключения к БД**

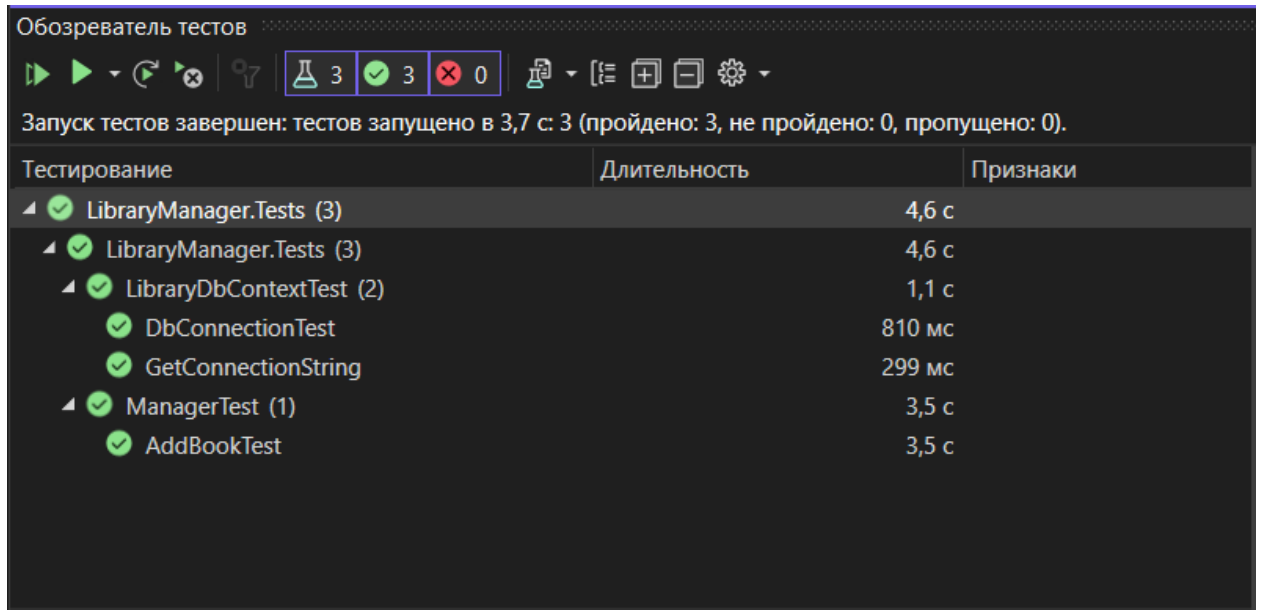
```
public void DbConnectionTest()  
{  
    var conStr =  
$"Host=locat=localhost;Database=postgres;Username=postgres;Password=00000";  
  
    var dbContext = new LibraryDbContext(conStr);  
    Assert.NotNull(dbContext);  
}
```

### **3) Тест добавления книги в БД**

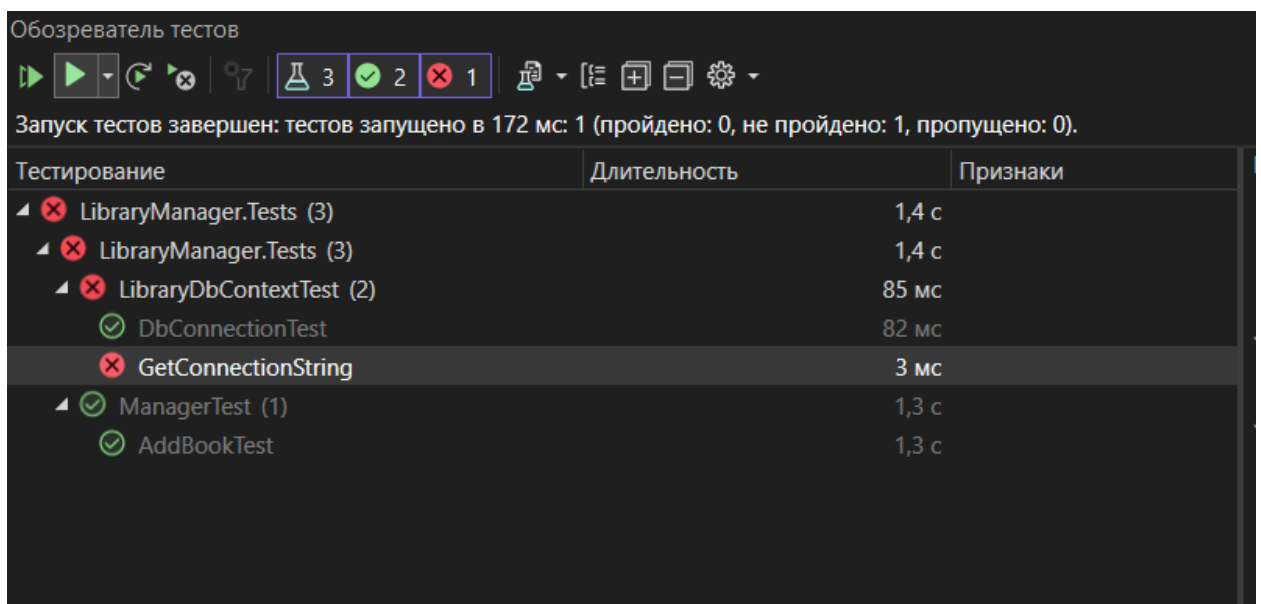
```
public void AddBookTest()  
{  
    var manager = new Manager();  
  
    int oldBooksCount = manager.Books.Count;  
  
    var newBook = new Book()  
    {  
        AuthorId = 1,  
        BookAuthor = manager.Authors[0],  
        Name = "TestBook",  
        Year = 1010  
    };  
    manager.AddBook(newBook);  
  
    int newBooksCount = manager.Books.Count;  
  
    Assert.Equal(oldBooksCount + 1, newBooksCount);  
}
```

## Скриншоты, иллюстрирующие выполнение тестов.

### 1) Удачное прохождение



### 2) Неудачное прохождение (GetConnectionString, из-за неправильного пути файла)



## Вывод

В ходе лабораторной работы была изучена библиотека xUnit. Написаны unit-тесты для некоторых функций (Добавление книги, получение информации из xml, подключение к БД)