МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тверской государственный технический университет»

(ТвГТУ)

Кафедра «Программного обеспечения»

**Лабораторная работа**

по дисциплине «Конструирование программных систем»

Тема: «Разработка UML модели проекта «Онлайн-библиотека»

Выполнил: студент группы

ПИН 17.06

Иванов Р.В.

Проверил:

Мальков А.А

Тверь 2020

# Введение

Цель работы: получить навык создания UML модели в системе Visual Studio 2013 и изучить особенности создания UML кода на языке C# .

# Ответы на контрольные вопросы

1. Какие этапы жизненного цикла разработки программного проекта поддерживает среда Visual Studio 2013.

* проектирование
* кодирование (программирование)
* тестирование и отладка

1. Какие UML-диаграммы можно построить с использованием Visual Studio 2013.

* Диаграмма классов
* Последовательности
* Вариантов использования
* Деятельности
* Компонентов

1. Каковы возможности автоматической генерации кода в Visual Studio 2013.

* Выбор шаблона для генерации кода (класс, перечисление, интерфейс, структура)
* Выбор директории для создаваемых классов

1. Рефакторинг в Visual Studio 2013. Его возможности и ограничения.

Возможности:

* Переименование
* Создание метода для выделенного кода
* Изменение порядка аргументов функции
* Извлечение интерфейса из определенного класса
* Инкапсулирование поля

# Диаграммы

## Последовательности



## Использования



## Активности



## Классов



### **Бизнес слой** - (Logical diagram)



### **Дата слой** - (Logical diagram*)*



### **Классы контролеры** - *(Logical diagram)*



### **Автор** - (Logical diagram)



### **Админ** - (Logical diagram)



### **Вход** - (Logical diagram)



### **Закладка** - (Logical diagram)



### **Книга** - (Logical diagram)



### **Поиск** - (Logical diagram)



### **Фильтр** - (Logical diagram)



### **Классы с ошибками** - *(Logical diagram)*



### **Модельные классы** - *(Logical diagram)*



## Пример доработанного сгенерированого кода

@XmlType(name = "book")  
@XmlRootElement  
public class Book extends Entity  
{  
 @XmlAttribute  
 private long bookId;  
 @XmlAttribute  
 private String bookName;  
 @XmlAttribute  
 private int releaseYear;  
 @XmlAttribute  
 private int pageCount;  
 @XmlAttribute  
 private String ISBN;  
 @XmlAttribute  
 private String publisher;  
 @XmlElement(type = Author.class)  
 private Author author;  
 @XmlAttribute  
 private boolean isTaken;  
  
 public Book(long bookId, String bookName, int releaseYear, int pageCount, String ISBN, String publisher, Author author ,boolean isTaken) {  
 this.bookId = bookId;  
 this.bookName = bookName;  
 this.releaseYear = releaseYear;  
 this.pageCount = pageCount;  
 this.ISBN = ISBN;  
 this.publisher = publisher;  
 this.author = author;  
 }  
  
 public Book(String bookName, int releaseYear, int pageCount, String ISBN, String publisher, Author author,boolean isTaken) {  
 this.bookName = bookName;  
 this.releaseYear = releaseYear;  
 this.pageCount = pageCount;  
 this.ISBN = ISBN;  
 this.publisher = publisher;  
 this.author = author;  
 }  
  
 public Book() {  
 }  
  
 public boolean getIsTaken() {  
 return isTaken;  
 }  
  
 public long getBookId() {  
 return bookId;  
 }  
  
 public String getBookName() {  
 return bookName;  
 }  
  
 public int getReleaseYear() {  
 return releaseYear;  
 }  
  
 public int getPageCount() {  
 return pageCount;  
 }  
  
 public String getISBN() {  
 return ISBN;  
 }  
  
 public String getPublisher() {  
 return publisher;  
 }  
  
 public Author getAuthor() {  
 return author;  
 }  
  
  
 @Override  
 public String toString() {  
 return "Book{" +  
 "bookId=" + bookId +  
 ", bookName='" + bookName +  
 ", releaseYear=" + releaseYear +  
 ", pageCount=" + pageCount +  
 ", ISBN='" + ISBN +  
 ", publisher='" + publisher +  
 ", authorId=" + author.getAuthorId() +  
 '}';  
 }