

**Московский государственный технический университет им.Н.Э.Баумана  
кафедра "Системы обработки информации и управления"**



## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к лабораторной работе №3

### **Средства составления проектной документации**

**«Технологии разработки программного обеспечения»**

Инструктор : Мария Валерьевна, Elizaveta Eliseeva

Email:2623859464@qq.com

Студент: Ван Чаочао

группа ИУ5-12М

2021/10/22

Учебно-методические материалы «Средства составления проектной документации» представляют собой методические указания к лабораторной работе по дисциплинам «Технологии разработки программного обеспечения» (по направлению магистерской подготовки) и «Технологии проектирования» (по направлению инженерной подготовки).

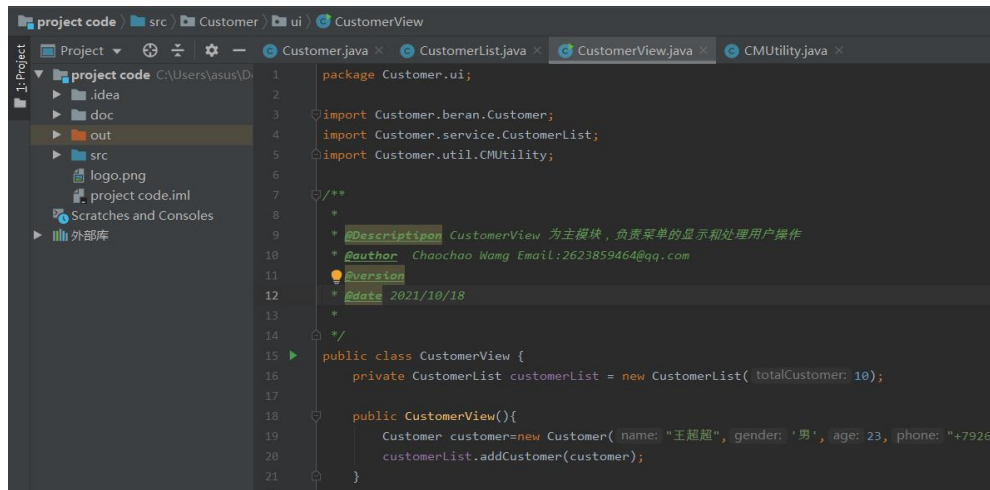
### **Цель работы:**

Изучить базовые возможности и получить навыки создания документации на основе исходного кода программ с применением CASE средств.

В материалах рассмотрены принципы составления проектной документации на основе исходного кода программ. Приведены инструкции по работе и команды документирования исходных кодов для автоматического составления документации в среде Doxygen. Рассмотрены примеры использования указанных средств. В заключительной части методических указаний приведены контрольные вопросы, список рекомендуемой литературы и пример задания.

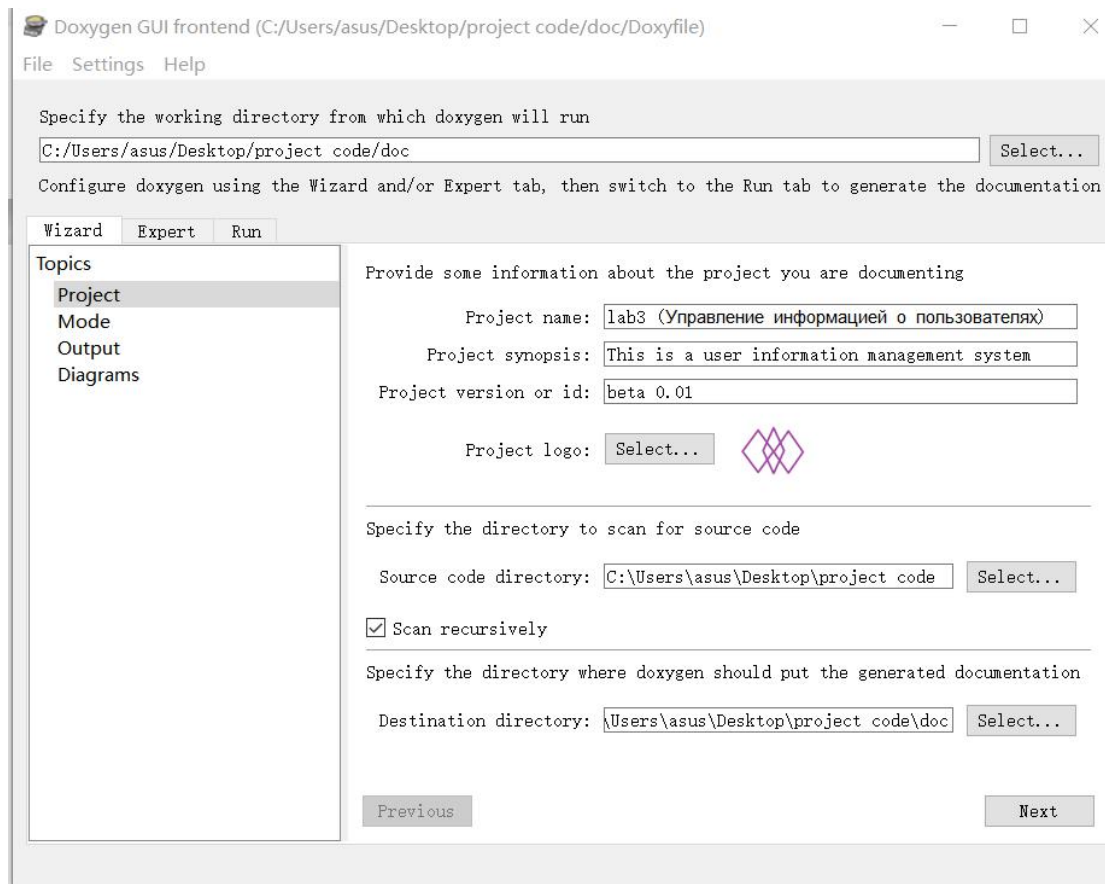
Ознакомившись с методическими указаниями и разобрав приведенные в нем примеры, студент может получить у преподавателя свой вариант задания и приступить к его выполнению.

### **Шаг 1: Создать проект**

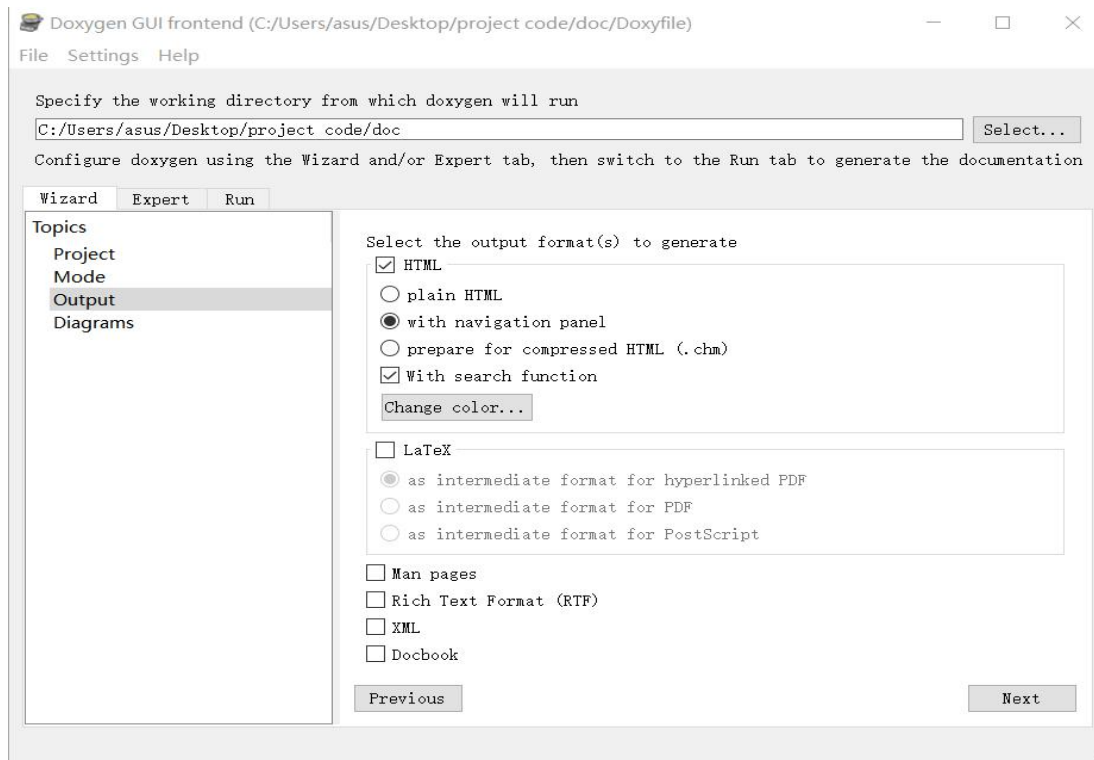


## Шаг2: Использование средств автоматического документирования

### 1: путь к создаваемой документации

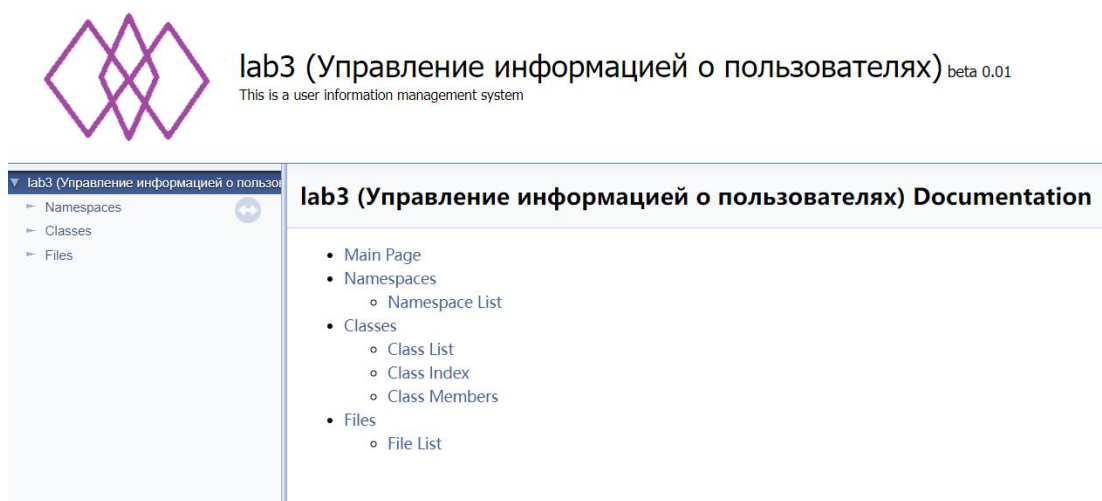


### 2: Указать на вкладке Output:



## Шаг3:Примеры результатов документирования

1: Страница проекта:



2:Страница классов:



lab3 (Управление информацией о пользователях) beta 0.01

Namespaces

Classes

Class List

Class Index

Class Members

Files

### Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Customer	
beran	
Customer	@Descriptipon Customer 为实体对象，用来封装客户信息
service	
CustomerList	@Descriptipon CustomerList 为Customer对象的管理模块，内部用数组管理一组Customer对象，并提供相应的添加、修改、删除和遍历方法
ui	
CustomerView	@Descriptipon CustomerView 为主模块，负责菜单的显示和处理用户操作
util	
CMUtility	CMUtility工具类：将不同的功能封装为方法，就是可以直接通过调用方法使用他的功能，二无需考虑具体的功能和实现细节。

### 3:Класс Персоны (фрагмент):



beran

Customer

Customer

getAge

getEmail

getGender

getName

getPhone

setAge

setEmail

setGender

setName

setPhone

age

email

gender

name

phone

service

ui

util

### Customer.beran.Customer Class Reference

Public Member Functions | Private Attributes | List of all members

@Descriptipon Customer 为实体对象，用来封装客户信息 More...

#### Public Member Functions

String	getName ()
void	setName (String name)
char	getGender ()
void	setGender (char gender)
int	getAge ()
void	setAge (int age)
String	getPhone ()
void	setPhone (String phone)
String	getEmail ()
void	setEmail (String email)
	Customer (String name, char gender, int age, String phone, String email)

#### Private Attributes

### 4:комментарии исходного кода и полученный результат для Doxygen.

Часть кода:

```
package Customer.ui;

import Customer.beran.Customer;
import Customer.service.CustomerList;
import Customer.util.CMUtility;

/**
 *
 * @Descriptipon CustomerView 为主模块，负责菜单的显示和处理用户操作
 * @author Chaochao Wang Email:2623859464@qq.com
 * @version
 * @date 2021/10/18
 *
 */
```

```

*/
public class CustomerView {
    private CustomerList customerList = new CustomerList(10);

    public CustomerView(){

        Customer customer=new Customer("王超超",'男
',23,"+79269525693","2623859464@qqw.com");
        customerList.addCustomer(customer);
    }

    /**
     * 显示客户信息管理软件的界面的方法
     */
    public void enterMainMenu(){

        boolean isFlag=true;
        while(isFlag) {

            System.out.println("\n-----客户信息管理软件-----");

            System.out.println("-----1 :添加客户-----");

            System.out.println("-----2 :修改客户-----");

            System.out.println("-----3 :删除客户-----");

            System.out.println("-----4 :客户列表-----");

            System.out.println("-----5 :退    出\n");

            System.out.println("                请选择 ( 1-5 ) :");

            char menu=CMUtility.readMenuSelection();
            switch (menu){
                case '1':
                    addNewCustomer();
                    break;
                case '2':
                    modifyCustomer();
                    break;
                case '3':
                    deleteCustomer();
                    break;
            }
        }
    }
}

```

```

        case '4':
            listAllCustomers();
            break;
        case '5':

            // System.out.println("退出");

            System.out.print("确认是否退出 ( Y/N):");

            char isExit=CMUtility.readConfirmSelection();
            if(isExit=='Y'){
                isFlag=false;
            }
            //break;
        }
    }
}

/**
 * 添加客户的操作
 */
private void addNewCustomer(){

    System.out.println("-----添加客户
-----");

    System.out.print("姓名 : ");
    String name=CMUtility.readString(10);

    System.out.print("性别 : ");
    char gender=CMUtility.readChar();

    System.out.print("年龄 : ");
    int age=CMUtility.readInt();

    System.out.print("电话 : ");
    String phone=CMUtility.readString(13);

    System.out.print("邮箱 : ");
    String email=CMUtility.readString(30);

    //将上述数据封装到对象中

```

```

        Customer customer=new Customer(name,gender,age,phone,email);
        boolean isSuccess=customerList.addCustomer(customer);

        if(isSuccess){

            System.out.println("-----添加完成
-----");
        }else{

            System.out.println("-----客户目录已满，添加失败
-----");
        }

    }
    /**
     * 修改客户的操作
     */
    private void modifyCustomer(){

        System.out.println("-----修改客户
-----");

        Customer cust;
        int number;
        for(;;){

            System.out.print("请选择待修改客户编号 ( -1 退出 ) ");

            number=CMUtility.readInt();
            if(number==1){
                return;
            }
            cust=customerList.getCustomer(number-1);
            if(cust==null){

                System.out.println("无法找到指定的客户！");

            }else{//找到了客户

                break;
            }
        }

        //修改客户信息

        System.out.print("姓名("+cust.getName()+"):");

```



```

String name=CMUtility.readString(10,cust.getName());

System.out.print("性别("+cust.getGender()+"):");

char gender=CMUtility.readChar(cust.getGender());

System.out.print("年龄("+cust.getAge()+"):");

int age=CMUtility.readInt(cust.getAge());

System.out.print("电话("+cust.getPhone()+"):");

String phone=CMUtility.readString(13,cust.getPhone());

System.out.print("邮箱("+cust.getEmail()+"):");

String email=CMUtility.readString(30,cust.getEmail());

Customer newCust=new Customer(name,gender,age,phone,email);

boolean isRepalaced=customerList.replaceCustomer(number-1,newCust);
if(isRepalaced){

    System.out.println("-----修改完成
-----");
}
else{

    System.out.println("-----修改失败
-----");
}
}
/**
 * 删除客户的操作
 */
private void deleteCustomer(){

    System.out.println("-----删除客户
-----");

    int number;
    for(;;){

        System.out.print("请选择待删除客户编号 ( -1 退出 ) ");

        number=CMUtility.readInt();
        if(number==-1){
            return;
        }
    }
}

```

```

        Customer customer=customerList.getCustomer(number-1);
        if(customer==null){

            System.out.println("无法找到指定客户");

        }else{
            break;
        }
    }

    //找到了指定客户

    System.out.print("确认是否删除 ( Y/N ) : ");

    char isDelete=CMUtility.readConfirmSelection();
    if(isDelete=='Y') {
        boolean deleteSuccess = customerList.deleteCustomer((number-1));

        if (deleteSuccess) {

            System.out.println("-----删除完成
            -----");
        } else {

            System.out.println("-----删除失败
            -----");
        }
    }else{
        return;
    }
}

/**
 * 显示客户列表的操作
 */
private void listAllCustomers(){

    System.out.println("-----客户列表
    -----");

    int total = customerList.getTotal();
    if(total==0){

        System.out.println("没有客户记录");

    }else{

```

```

        System.out.println("编号 姓名 性别 年龄\t\t电话\t\t\t\t\t邮箱");

        Customer[] custs=customerList.getAllCustomer();
        for(int i=0;i<custs.length;i++){
            Customer cust =custs[i];

System.out.println((i+1)+"\t"+cust.getName()+"\t"+cust.getGender()+"\t"+cust.ge
tAge()+"\t"+cust.getPhone()+"\t"+cust.getEmail());
        }
    }

    System.out.println("-----客户列表完成
-----");
}

    public static void main(String[] args) {
        CustomerView view=new CustomerView();
        view.enterMainMenu();
    }
}

```