ГУАП

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Н.А. Соловьева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙСИСТЕМЫ. РАЗРАБОТКА ДИАГРАММЫ ВАРИАНТОВИСПОЛЬЗОВАНИЯ |
| по курсу: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4941 |  |  |  | Н.С. Горбунов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**Цель работы:**

Научиться автоматической генерации кода программ ИС (генерация кода по диаграмме классов). Изучить обратное проектирование (генерация диаграммы классов по коду).

**Задание:** Гостиничный комплекс

Листинг классов

namespace Model1

{

class Admin : Employee

{

private bool addEmloyee ()

{

}

}

}

namespace Model1

{

class Category

{

private int ID;

private string name;

private string description;

//

private Service Services;

Service 1..\*;

public List<ServicePackage> getAll ()

{

}

}

}

namespace Model1

{

class Customer : SystemUser, SystemUser

{

private string adress;

//

private ShoppingCart Cart;

public bool register ()

{

}

public object search ()

{

}

public bool login ()

{

}

}

}

namespace Model1

{

//

class Department

{

private int ID;

private string name;

private string description;

//

private Employee Employes;

}

}

namespace Model1

{

abstract class Employee : SystemUser, SystemUser

{

public bool isFree ()

{

}

public bool giveHoliday ()

{

}

public bool changeSchedule ()

{

}

public bool changeDepartment ()

{

}

public bool reportOfErr ()

{

}

}

}

namespace Model1

{

class ErrorReport

{

private int id;

private string name;

private string description;

private ReportStatus status;

}

}

namespace Model1

{

class Manager : Employee

{

public bool addService ()

{

}

public bool editService ()

{

}

public bool deleteService ()

{

}

public bool addPromotion ()

{

}

public bool editPromotion ()

{

}

public bool deletePromotion ()

{

}

}

}

namespace Model1

{

//

class Order

{

private int orderID;

private DateTime dateCreated;

private DateTime dateExecution;

private OrderStatus status;

private PayStatus payStatus;

//

private ServicePackage ServicePackages;

//

private Promotion promotion;

public string showDetails() ()

{

}

public Money calcOrderPrice ()

{

}

public bool callWorker ()

{

}

public bool addPromotion ()

{

}

public bool pay ()

{

}

}

}

namespace Model1

{

class Prametr

{

private int typeID;

private string value;

}

}

namespace Model1

{

class Promotion

{

private int promID;

private int discount;

private DateTime timeStart;

private DateTime timeEnd;

}

}

namespace Model1

{

class Schedule

{

//

private WorkerDay Days;

}

}

namespace Model1

{

class Service

{

public string name;

private Money price;

private string description;

//

private Prametr parameters;

Category 1..\*;

}

}

namespace Model1

{

class ServicePackage

{

private int ID;

private string name;

private string description;

//

private Service Services;

public bool addService ()

{

}

}

}

namespace Model1

{

class ShoppingCart

{

public int cartID;

private object date;

//

private Order orders;

public bool doAllOrder ()

{

}

public bool removeOrder ()

{

}

private Money calcTotal ()

{

}

}

}

namespace Model1

{

abstract class SystemUser

{

private object ID;

private string login;

private string password;

private string phone;

private string email;

public string getPhone ()

{

}

public bool checkAuth ()

{

}

}

}

namespace Model1

{

class Worker : Employee

{

public bool changeOrderStatus ()

{

}

}

}

namespace Model1

{

class WorkerDay

{

private Time timeStart;

private Time timeEnd;

private Day dayOfWeek;

}

}

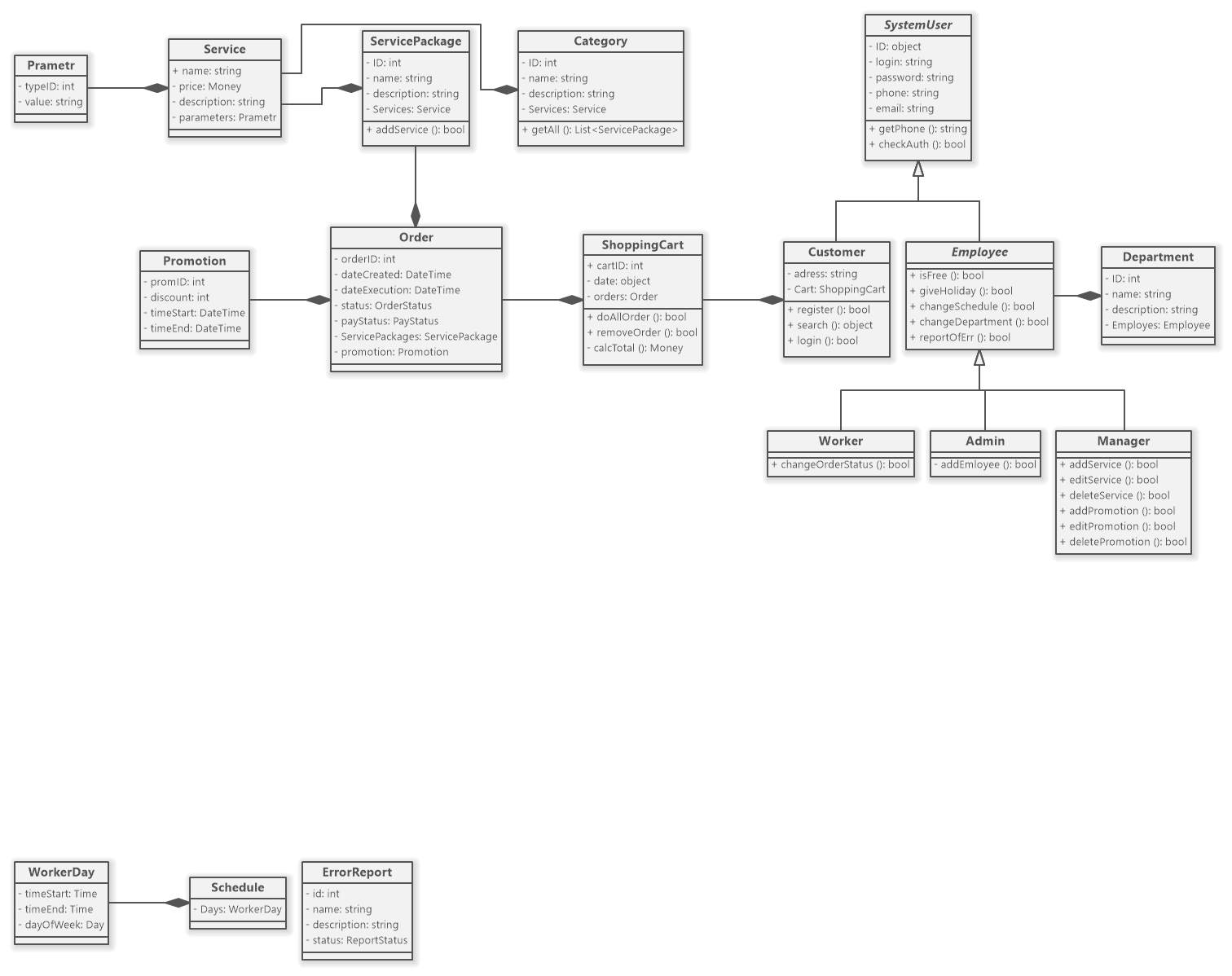


Рисунок 1. Сгенерированная диаграмма классов

Вывод:

Научился автоматической генерации кода программ ИС (генерация кода по диаграмме классов). Изучил обратное проектирование (генерация диаграммы классов по коду).

Список используемых источников:

1. Software Ideas Modeler [Электронный ресурс]: CASE tool for software design & analysis. – Электронные данные – Dušan Rodina, 2009-2021. – Режим доступа: https://www.softwareideas.net/

2. Иванова Г. С. Технология программирования: учебник для вузов, 2-е изд.: – М.: ИД КноРус, 2015. – 333 с.: ил.

3. SmartDraw – Data Flow Diagram [Электронный ресурс]. – Электронные данные – SmartDraw, 1994-2021. – Режим доступа: https://www.smartdraw.com/data-flow-diagram/

4. Visual Paradigm – Functional Modeling with Data Flow Diagram Tutorial [Электронный ресурс]. – Электронные данные – Visual Paradigm,

1999-2021. – Режим доступа: https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp