namespace SOLID

{

public class Employee

{

public int ID { get; set; }

public string FullName { get; set; }

/// <summary>

/// Данный метод добавляет в БД нового сотрудника

/// </summary>

/// <param name="em">Объект для вставки</param>

/// <returns>Результат вставки новых данных</returns>

public bool Add(Employee emp)

{

// Вставить данные сотрудника в таблицу БД

return true;

}

}

public class EmployeeReport

{

/// <summary>

/// Отчет по сотруднику

/// </summary>

public void GenerateReport(Employee em)

{

// Генерация отчета по деятельности сотрудника

}

}

}

public class EmployeeReport

{

// <summary>

/// Тип отчета

/// </summary>

public string TypeReport { get; set; }

/// <summary>

/// Отчет по сотруднику

/// </summary>

public void GenerateReport(Employee em)

{

if (TypeReport == "CSV")

{

// Генерация отчета в формате CSV

}

if (TypeReport == "PDF")

{

// Генерация отчета в формате PDF

}

}

}

public class IEmployeeReport

{

/// <summary>

/// Метод для создания отчета

/// </summary>

public virtual void GenerateReport(Employee em)

{

// Базовая реализация, которую нельзя модифицировать

}

}

public class EmployeeCSVReport : IEmployeeReport

{

public override void GenerateReport(Employee em)

{

// Генерация отчета в формате CSV

}

}

public class EmployeePDFReport : IEmployeeReport

{

public override void GenerateReport(Employee em)

{

// Генерация отчета в формате PDF

}

}

public interface IEmployee

{

string GetEmployeeDetails(int employeeId);

}

public interface IWork

{

string GetWorkDetails(int employeeId);

}

public class SeniorEmployee : IWork, IEmployee

{

public string GetWorkDetails(int employeeId)

{

return "Senior Work";

}

public string GetEmployeeDetails(int employeeId)

{

return "Senior Employee";

}

}

public class JuniorEmployee : IEmployee

{

public string GetEmployeeDetails(int employeeId)

{

return "Junior Employee";

}

}

public interface Payment

{

void PayWebMoney();

void PayCreditCard();

void PayPhoneNumber();

}

public interface WebMoneyPayment

{

void PayWebMoney();

}

public interface CreditCardPayment

{

void PayCreditCard();

}

public interface PhoneNumberPayment

{

void PayPhoneNumber();

}

internal class TerminalPaymentService : WebMoneyPayment, CreditCardPayment

{

public void PayWebMoney() { }

public void PayCreditCard() { }

}

internal class InternetPaymentService : WebMoneyPayment, CreditCardPayment, PhoneNumberPayment

{

public void PayWebMoney() { }

public void PayCreditCard() { }

public void PayPhoneNumber() { }

}

public interface IMessenger

{

void Send();

}

public class Email : IMessenger

{

public void Send()

{

// код для отправки email-письма

}

}

public class SMS : IMessenger

{

public void Send()

{

// код для отправки SMS

}

}

// Уведомление

public class Notification

{

private IMessenger \_messenger;

public Notification()

{

\_messenger = new Email();

}

public void DoNotify()

{

\_messenger.Send();

}

}