ФГАОУ ВО «Омский государственный технический университет» Кафедра «Автоматизированные системы обработки информации и управления»

### ОТЧЁТ

по дисциплине «Разработка программного кода» студента Чернуха Ильи Евгеньевича группы ИВТМ-221

на тему «Фабричный метод»

### Пояснительная записка

Шифр работы РГР–02068999–43–Ивтм-221 ПЗ

Направление 09.04.01 Информатика и вычислительная техника Направленность (профиль) «Автоматизированные системы обработки информации и управления»

Доцент П.Н. Надточий

Студент И.Е. Чернуха

Омск 2023

**Задание:**

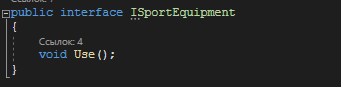
Приведите пример кода (классы\интерфейсы с комментариями) использующий какой-либо порождающий шаблон проектирования и опишите задачу, которую он решает. На примере рассмотрим фабричный метод.

Данная пример будет реализован с помощью языка C#. C# — объектно-ориентированный, **ориентированный на компоненты** язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — **объектно-ориентированный** язык.

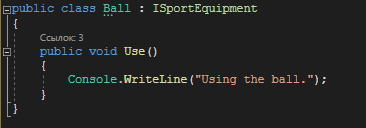
Фабричный метод (Factory Method) - это паттерн, который определяет интерфейс для создания объектов некоторого класса, но непосредственное решение о том, объект какого класса создавать происходит в подклассах. То есть паттерн предполагает, что базовый класс делегирует создание объектов классам-наследникам.

**Решение:**

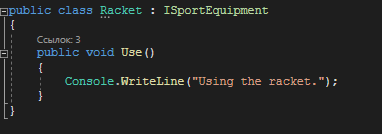
Пример кода на C#, который использует порождающий шаблон проектирования "Фабричный метод" для создания классов и интерфейсов, связанных со спортом.



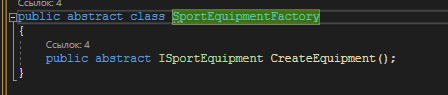
ISportEquipment: Интерфейс для спортивного снаряжения, содержащий метод Use(), который будет реализован каждым конкретным классом спортивного снаряжения.



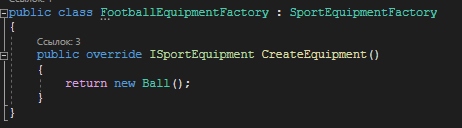
Ball: Реализация спортивного мяча, реализующая интерфейс ISportEquipment и определяющая метод Use(), который выводит сообщение "Using the ball.".



Racket: Реализация спортивной ракетки, также реализующая интерфейс ISportEquipment и имеющая метод Use(), который выводит сообщение "Using the racket.".

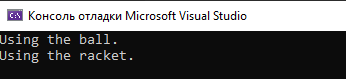


SportEquipmentFactory: Абстрактный класс фабрики спортивного снаряжения, определяющий абстрактный метод CreateEquipment(), который будет реализован каждой конкретной фабрикой.

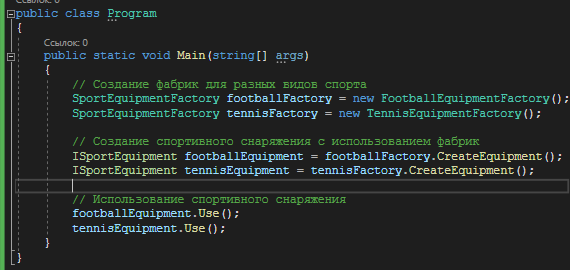


FootballEquipmentFactory: Реализация фабрики для создания футбольного снаряжения, наследующая от SportEquipmentFactory.

**Демонстрация работы:**



Я создал фабрики для разных видов спорта, создал снаряжение с использованием фабрики и использовал спортивное снаряжение.



**Листинг кода:**

using System;

// Интерфейс для спортивного снаряжения

public interface ISportEquipment

{

void Use();

}

// Реализация спортивного мяча

public class Ball : ISportEquipment

{

public void Use()

{

Console.WriteLine("Using the ball.");

}

}

// Реализация спортивной ракетки

public class Racket : ISportEquipment

{

public void Use()

{

Console.WriteLine("Using the racket.");

}

}

// Абстрактный класс фабрики спортивного снаряжения

public abstract class SportEquipmentFactory

{

public abstract ISportEquipment CreateEquipment();

}

// Реализация фабрики для создания футбольного снаряжения

public class FootballEquipmentFactory : SportEquipmentFactory

{

public override ISportEquipment CreateEquipment()

{

return new Ball();

}

}

// Реализация фабрики для создания теннисного снаряжения

public class TennisEquipmentFactory : SportEquipmentFactory

{

public override ISportEquipment CreateEquipment()

{

return new Racket();

}

}

public class Program

{

public static void Main(string[] args)

{

// Создание фабрик для разных видов спорта

SportEquipmentFactory footballFactory = new FootballEquipmentFactory();

SportEquipmentFactory tennisFactory = new TennisEquipmentFactory();

// Создание спортивного снаряжения с использованием фабрик

ISportEquipment footballEquipment = footballFactory.CreateEquipment();

ISportEquipment tennisEquipment = tennisFactory.CreateEquipment();

// Использование спортивного снаряжения

footballEquipment.Use();

tennisEquipment.Use();

}

}