

**Московский государственный технический  
университет им. Н.Э. Баумана**

**Факультет “Радиотехнический”  
Кафедра “Системы обработки информации и управления”**

**Курс «Парадигмы и конструкции языка»**

**Отчет по домашнему заданию**

Выполнил:  
студент группы РТ5-31Б:  
Зеленева И.Е.

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2025 г.

## Описание задания

Изучить принцип внедрения зависимостей и продемонстрировать его работу на примере программы на языке F#.

### Текст программы

```
[<AbstractClass>]
type CoffeeBrewer() =
    abstract member Brew: unit -> string
type EspressoBrewer() =
    inherit CoffeeBrewer()
    override this.Brew() = "Приготовлен эспрессо: насыщенный и крепкий"

type CappuccinoBrewer() =
    inherit CoffeeBrewer()
    override this.Brew() = "Приготовлен капучино: с молочной пенкой"

type LatteBrewer() =
    inherit CoffeeBrewer()
    override this.Brew() = "Приготовлен латте: много молока, мягкий вкус"

type AmericanoBrewer() =
    inherit CoffeeBrewer()
    override this.Brew() = "Приготовлен американо: эспрессо с водой"
[<AbstractClass>]
type CoffeeMachine() =
    abstract member MakeCoffee: unit -> string

type AutomaticCoffeeMachine(brewer: CoffeeBrewer) =
    inherit CoffeeMachine()

    override this.MakeCoffee() =
        $"Автоматическая кофемашина: {brewer.Brew()}"
type ManualCoffeeMachine(brewer: CoffeeBrewer) =
    inherit CoffeeMachine()

    override this.MakeCoffee() =
        $"Ручная кофемашина (бариста): {brewer.Brew()}"

type CoffeeService(machine: CoffeeMachine) =
    member this.ServeCoffee() =
        machine.MakeCoffee()
module CoffeeFactory =
    let createEspressoMachine() =
        AutomaticCoffeeMachine(EspressoBrewer())

    let createCappuccinoMachine() =
        AutomaticCoffeeMachine(CappuccinoBrewer())

    let createLatteMachine() =
        ManualCoffeeMachine(LatteBrewer())

    let createAmericanoMachine() =
        ManualCoffeeMachine(AmericanoBrewer())
module Program =
    let demonstrateCoffeePreparation() =
        printfn "=== Приготовление кофе с DI ==="
        let espressoService =
            CoffeeService(CoffeeFactory.createEspressoMachine())
        let cappuccinoService =
            CoffeeService(CoffeeFactory.createCappuccinoMachine())
        let latteService = CoffeeService(CoffeeFactory.createLatteMachine())
```

```

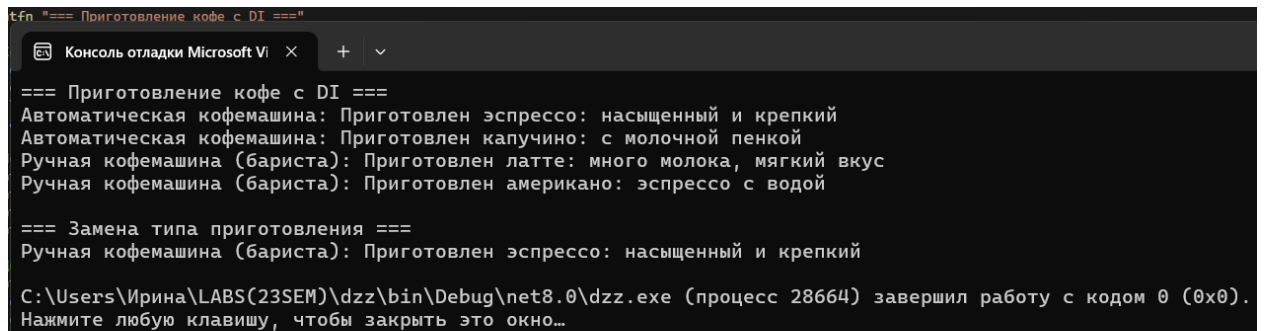
let americanoService =
CoffeeService(CoffeeFactory.createAmericanoMachine())

printfn "%s" (espressoService.ServeCoffee())
printfn "%s" (cappuccinoService.ServeCoffee())
printfn "%s" (latteService.ServeCoffee())
printfn "%s" (americanoService.ServeCoffee())

printfn "\n=== Замена типа приготовления ==="
let customMachine = ManualCoffeeMachine(EspressoBrewer())
let customService = CoffeeService(customMachine)
printfn "%s" (customService.ServeCoffee())

[<EntryPoint>]
let main argv =
    demonstrateCoffeePreparation()
    0

```



The screenshot shows a Visual Studio debug console window titled "Консоль отладки Microsoft Vi". The output of the program is as follows:

```

tfn "=== Приготовление кофе с DI ==="

=== Приготовление кофе с DI ===
Автоматическая кофемашина: Приготовлен эспрессо: насыщенный и крепкий
Автоматическая кофемашина: Приготовлен капучино: с молочной пенкой
Ручная кофемашина (бариста): Приготовлен латте: много молока, мягкий вкус
Ручная кофемашина (бариста): Приготовлен американо: эспрессо с водой

=== Замена типа приготовления ===
Ручная кофемашина (бариста): Приготовлен эспрессо: насыщенный и крепкий

C:\Users\Ирина\LABS(23SEM)\dzz\bin\Debug\net8.0\dzz.exe (процесс 28664) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно...

```