

Божко Роман Вячеславович, ШІ-1

## Формальна специфікація системи “Кавовий автомат”

Було застосовано модельно-орієнтований метод специфікації, де аспекти системи описано за допомогою аксіом, характерних для специфікацій, орієнтованих на властивості.

Було обрано поведінковий тип специфікації, який передбачає опис обмежень на поведінку об'єкта — у цьому випадку кавового автомата. Опис охоплює функціональні можливості та виконання. Водночас цю специфікацію частково можна віднести до структурного типу, оскільки в ній також відображено внутрішню структуру об'єкта, зокрема напій, оплату тощо.

## Оновлена формальна специфікація

Формальна специфікація згідно з [цією презентацією](#):

```
sort CoffeeMachine
imports INTEGER, BOOLEAN
```

ChooseDrink - обирає напій у кавовому автоматі

PayForDrink - здійснює оплату за обраний напій

PrepareDrink - готує напій після оплати

DispenseDrink - видає приготований напій

Create - створює новий кавовий автомат у вимкненому стані

TurnOn - вмикає кавовий автомат

TurnOff - вимикає кавовий автомат

HasIngredients - перевіряє наявність інгредієнтів для напою

HasPayment - перевіряє, чи була здійснена оплата

DrinkReady - перевіряє, чи напій готовий

ChooseDrink (CoffeeMachine, Drink) → CoffeeMachine

PayForDrink (CoffeeMachine) → CoffeeMachine

PrepareDrink (CoffeeMachine) → CoffeeMachine

DispenseDrink (CoffeeMachine) → CoffeeMachine

Create → CoffeeMachine

TurnOn (CoffeeMachine) → CoffeeMachine

TurnOff (CoffeeMachine) → CoffeeMachine

HasIngredients (CoffeeMachine, Drink) → Boolean

```

HasPayment (CoffeeMachine) → Boolean
DrinkReady (CoffeeMachine) → Boolean

ChooseDrink (CM, D) =
  if CM = Create or not TurnOn (CM) then CM exception (Machine is off)
  elseif DrinkReady (CM) then CM exception (Drink is already being prepared)
  else CM' -- встановлює вибраний напій

PayForDrink (CM) =
  if not ChooseDrink (CM, D) then CM exception (No drink selected)
  elseif not HasPayment (CM) then CM exception (Payment terminal error)
  else CM' -- здійснює оплату

PrepareDrink (CM) =
  if not PayForDrink (CM) then CM exception (Payment not completed)
  elseif not HasIngredients (CM, D) then CM exception (Not enough
ingredients)
  else CM' -- починає приготування напою

DispenseDrink (CM) =
  if not PrepareDrink (CM) then CM exception (Drink not prepared)
  else CM' -- видає готовий напій

HasIngredients (Create, D) = false
HasIngredients (TurnOn (CM), D) = true -- передбачається, що інгредієнти є
при ввімкненні

HasPayment (Create) = false
HasPayment (PayForDrink (CM)) = true

DrinkReady (Create) = false
DrinkReady (PrepareDrink (CM)) = true

```

## Реалізація функцій

Перейдемо до реалізації описаних функцій за допомогою Dafny в середовищі VS Code.

### Змінні

Реалізуємо згадані функції в класі `CoffeeMachine`, де також зберігається інформація про стан кавового автомату\платіжного терміналу, наявні інгредієнти, загальну кількість приготованих напоїв, загальну суму оплат а також інші змінні стану машини.

```
class CoffeeMachine {  
    var isOn: bool  
    var isPreparingDrink: bool  
    var selectedDrink: string  
    var isPaid: bool  
    var isDrinkPrepared: bool  
    var totalIncome: int  
    var totalDrinksMade: int  
  
    var coffeeAmount: int  
    var milkAmount: int  
    var waterAmount: int  
    var sugarAmount: int  
  
    var isTerminalWorking: bool
```

## Конструктор

Клас `CoffeeMachine` крім цільових методів та допоміжних функцій також містить конструктор, де задаються початкові значення наведених вище змінних, якщо значення задовольняють обмеження, наприклад `initialMilk`, `initialSugar`.

```

constructor(initialCoffee: int, initialMilk: int, initialWater: int, initialSugar: int)
  requires initialCoffee >= 0
  requires initialMilk >= 0
  requires initialWater >= 0
  requires initialSugar >= 0
  ensures coffeeAmount == initialCoffee
  ensures milkAmount == initialMilk
  ensures waterAmount == initialWater
  ensures sugarAmount == initialSugar
  ensures isOn == false
  ensures isPreparingDrink == false
  ensures selectedDrink == ""
  ensures isPaid == false
  ensures isDrinkPrepared == false
  ensures totalIncome == 0
  ensures totalDrinksMade == 0
  ensures isTerminalWorking == true
{
  coffeeAmount := initialCoffee;
  milkAmount := initialMilk;
  waterAmount := initialWater;
  sugarAmount := initialSugar;
  isOn := false;
  isPreparingDrink := false;
  selectedDrink := "";
  isPaid := false;
  isDrinkPrepared := false;
  totalIncome := 0;
  totalDrinksMade := 0;
  isTerminalWorking := true;
}

```

**Функція: Обрати напій**

```

method SelectDrink(drinkName: string) returns (success: bool)
  modifies this
  ensures !old(isOn) ==> !success
  ensures old(isPreparingDrink) ==> !success
  ensures success ==> selectedDrink == drinkName
{
  if (!isOn || isPreparingDrink) {
    success := false;
    return success;
  }

  var hasIngredients := HasEnoughIngredients(drinkName);
  if (!hasIngredients) {
    success := false;
    return success;
  }

  selectedDrink := drinkName;
  success := true;
  return success;
}

```

**Функція: Оплатити напій**

```

method PayForDrink(drinkPrice: int) returns (success: bool)
    modifies this
    requires drinkPrice > 0
    ensures selectedDrink == "" ==> !success
    ensures !isTerminalWorking ==> !success
    ensures success ==> isPaid
    ensures success ==> totalIncome == old(totalIncome) + drinkPrice
    ensures success ==> totalDrinksMade == old(totalDrinksMade) + 1
{
    if (selectedDrink == "") {
        // Неможливо оплатити, якщо не вибрано жодного напою
        success := false;
        return success;
    }

    if (!isTerminalWorking) {
        // Неможливо оплатити, якщо термінал не працює
        success := false;
        return success;
    }

    isPaid := true;
    totalIncome := totalIncome + drinkPrice;
    totalDrinksMade := totalDrinksMade + 1;
    success := true;
    return success;
}

```

**Функція: Приготувати напій**

```
method PrepareDrink() returns (success: bool)
  modifies this
  ensures !isPaid ==> !success
  ensures success ==> isDrinkPrepared
  ensures success ==> isPreparingDrink
{
  if (!isPaid) {
    success := false;
    return success;
  }

  isPreparingDrink := true;
  isDrinkPrepared := true;
  success := true;
  return success;
}
```

**Функція: Видати напій**

```

method DispenseDrink() returns (success: bool)
    modifies this
    ensures !isDrinkPrepared ==> !success
    ensures success ==> !isDrinkPrepared
    ensures success ==> !isPreparingDrink
    ensures success ==> !isPaid
    ensures success ==> selectedDrink == ""

    if (!isDrinkPrepared) {
        // Не можна видати напій, якщо він не готовий
        success := false;
        return success;
    }

    // Оновлення кількості інгредієнтів залежно від приготованого напою
    if (selectedDrink == "espresso") {
        coffeeAmount := coffeeAmount - 7;
        waterAmount := waterAmount - 30;
    } else if (selectedDrink == "cappuccino") {
        coffeeAmount := coffeeAmount - 7;
        milkAmount := milkAmount - 70;
        waterAmount := waterAmount - 30;
    } else if (selectedDrink == "latte") {
        coffeeAmount := coffeeAmount - 7;
        milkAmount := milkAmount - 120;
        waterAmount := waterAmount - 30;
    } else if (selectedDrink == "americano") {
        coffeeAmount := coffeeAmount - 7;
        waterAmount := waterAmount - 100;
    }

    // Скинути стан машини для наступного замовлення
    isDrinkPrepared := false;
    isPreparingDrink := false;
    isPaid := false;
    selectedDrink := "";

    success := true;
    return success;

```

Для проведення симуляції поведінки цієї системи створимо окрему функцію Main, що буде вхідною точкою для нашого проєкту.



```

method Main()
{
    var machine := new CoffeeMachine(500, 1000, 2000, 300);

    machine.PowerToggle();
    print "Machine powered on\n";

    var selectionSuccess := machine.SelectDrink("cappuccino");
    if (selectionSuccess) {
        print "Selected drink: cappuccino\n";
    } else {
        print "Failed to select drink\n";
    }

    var paymentSuccess := machine.PayForDrink(35);
    if (paymentSuccess) {
        print "Payment successful. Total income: ", machine.totalIncome, "\n";
    } else {
        print "Payment failed\n";
    }

    var preparationSuccess := machine.PrepareDrink();
    if (preparationSuccess) {
        print "Drink preparation started\n";
    } else {
        print "Failed to prepare drink\n";
    }

    var dispensingSuccess := machine.DispenseDrink();
    if (dispensingSuccess) {
        print "Drink dispensed. Remaining ingredients:\n";
        print "Coffee: ", machine.coffeeAmount, "g\n";
        print "Milk: ", machine.milkAmount, "ml\n";
        print "Water: ", machine.waterAmount, "ml\n";
        print "Sugar: ", machine.sugarAmount, "g\n";
    } else {
        print "Failed to dispense drink\n";
    }
}

```

У реалізованому вище методі **Main**:

- Створюємо кавовий апарат, в який передаємо початкові значення: 500 г кави, 1000 мл молока, 2000 мл води та 300 г цукру.
- Включаємо кавовий апарат за допомогою методу **PowerToggle** та виводимо повідомлення про успішне включення.
- Обираємо напій "cappuccino" за допомогою методу **SelectDrink** та перевіряємо, чи вибір був успішним. Якщо так, виводимо повідомлення з назвою обраного напою, інакше - повідомлення про невдачу.

- Оплачуємо напій сумою 35 у.о. за допомогою методу `PayForDrink` та перевіряємо успішність оплати. Якщо оплата пройшла успішно, виводимо повідомлення з загальним доходом кавового апарату, інакше - повідомлення про відмову.
- Починаємо приготування напою за допомогою методу `PrepareDrink` та перевіряємо успішність початку приготування. Виводимо відповідне повідомлення залежно від результату.
- Видаємо напій за допомогою методу `DispenseDrink` та перевіряємо успішність. У разі успішної видачі виводимо інформацію про залишок інгредієнтів у кавовому апараті (кава, молоко, вода, цукор), інакше - повідомлення про невдачу.
- Кожен крок цього методу супроводжується відповідним `print`'ом, що дозволяє переконатись, що програма працює як слід.

Ось результат роботи даної програми:

```
Machine powered on
Selected drink: cappuccino
Payment successful. Total income: 35
Drink preparation started
Drink dispensed. Remaining ingredients:
Coffee: 493g
Milk: 930ml
Water: 1970ml
Sugar: 300g

Total drinks made: 1
Total income: 35
```

## Висновок

Перша контрольна робота дала нам змогу розібратися з різними видами та способами формальної специфікації, навчитися застосовувати ці знання на практиці для опису ключових функцій систем, а також набути досвід роботи з мовою Dafny у VS Code.