МИНИСТЕРСТВО НАУКИ И ВЫШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРТСВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**Кафедра вычислительной техники**



**Лабораторная работа №4**

**по предмету «Инжиниринг программного обеспечения»**

Графический интерфейс с динамической разметкой

Вариант 1

Группа: АВТ-143

Студенты: Чалов В.П., Комаров П.В.

Преподаватель: Романов Е. Л.

Новосибирск 2024

Содержание:

[Результат: 2](#_Toc176730157)

[Задача: 3](#_Toc176730158)

[Описание: 3](#_Toc176730159)

[UML диаграмма: 3](#_Toc176730160)

[Интерфейс: 4](#_Toc176730161)

[Устройство приложения изнутри: 6](#_Toc176730162)

[Известные проблемы устройства приложения: 11](#_Toc176730163)

[Вывод: 11](#_Toc176730164)

Результат:[Ссылка на приложение](https://github.com/peachyhead/pi_4/tree/master)

# Задача:

Физический объект – кастрюля на плите: текущее положение конфорки, температура и объем содержимого, состояние нагревания и кипения, теплоотдача при открытой и закрытой крышке.

# Описание:

Класс-контроллер реализует в отдельном потоке в реальном времени модель поведения физического объекта. Классы GUI (формы) получают ссылку на объект-контроллер с интерфейсом команд и подписываются на получение от него событий через соответствующий интерфейс (шаблон наблюдатель). При появлении в контроллере события последний вызывает соответствующий метод во всех подписанных объектах. При закрытии объекта класса GUI он отписывается от контроллера. События и команды в интерфейсах должны обеспечивать объектам внешнего представления полный набор возможностей управления и отображения состояния физического объекта.

# UML диаграмма:

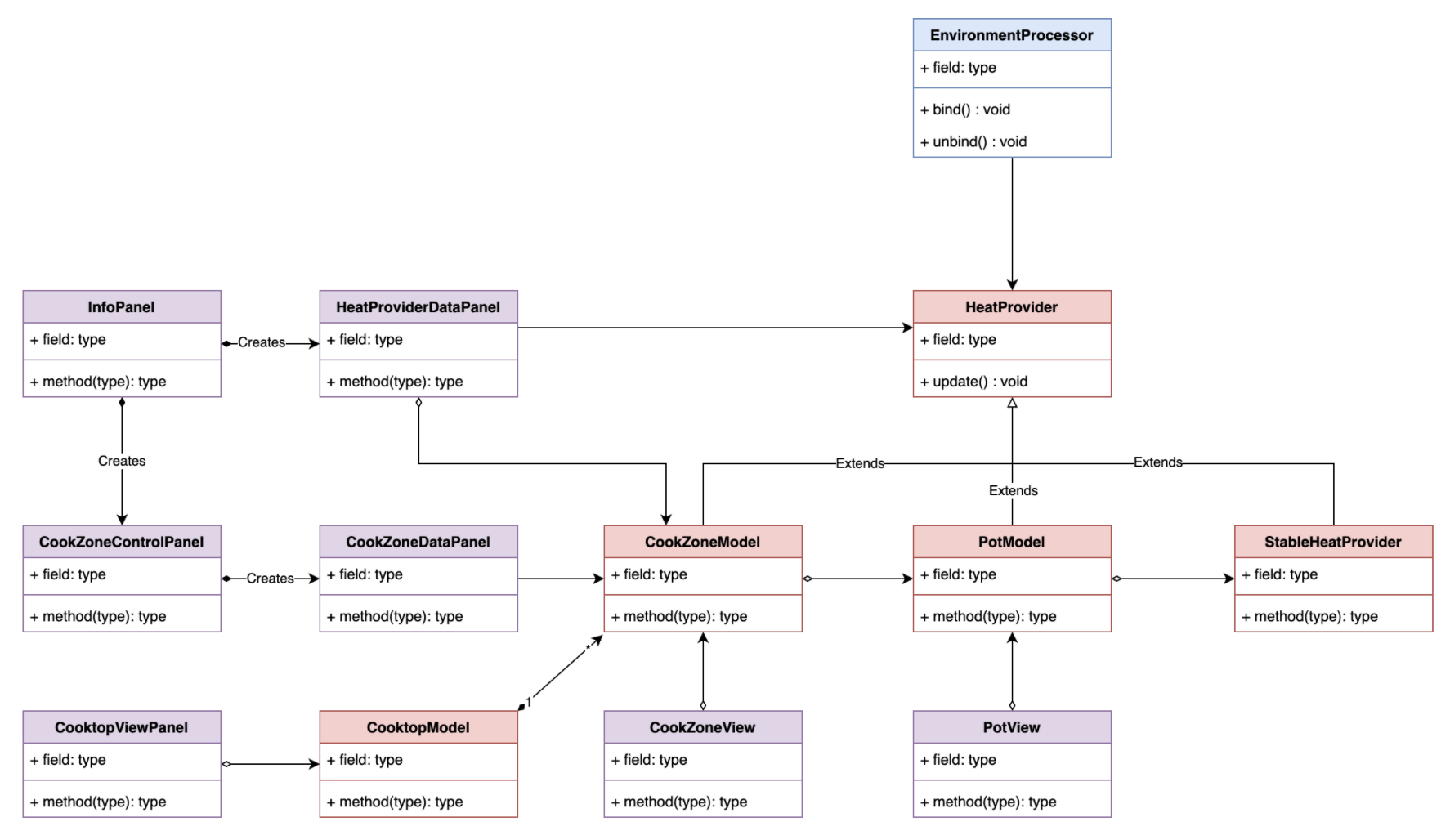


Рисунок 1. Поверхностная uml диаграмма приложения.

На диаграмме вынесен жизненный цикл в отдельный модуль (каковым он не является), чтобы улучшить читаемость.

*Определение сущностей диаграммы:*

**красный** - модель данных

**фиолетовый** – отображение

**голубой** - сервис

Не станем упоминать на диаграмме вспомогательные инструменты (SignalBus, SignalListener). Можно назвать это сахаром, который делает вещи удобнее.

## Моделирование

Для создания необходимой симуляции, нужно учитывать:

- передачу тепла от конфорки к кастрюле

- передачу тепла от кастрюли к ее содержимому

- теплообмен с окружающей средой

- состояния нагрева жидкости

Мы будем использовать следующие принципы термодинамики:

**1. Закон теплопередачи (уравнение теплопередачи)**

Теплопередача между двумя объектами или между объектом и окружающей средой рассчитывается по формуле:

Где:

• Q — количество тепла, переданное между объектом и окружающей средой (Дж).

• k — коэффициент теплопередачи (Вт/м²°C).

• S — площадь поверхности теплопередачи (м²).

•  — разность температур между объектом и окружающей средой.

**2. Изменение температуры в результате поглощенного тепла**

Температура объекта меняется в зависимости от того, сколько тепла он поглощает. Это рассчитывается по формуле:

Где:

• — изменение температуры объекта (°C).

• Q — количество тепла, поглощенное объектом (Дж).

• C — удельная теплоемкость объекта (Дж/кг°C).

• m — масса объекта (кг).

Температура объекта обновляется по формуле:

**3. Масса объекта**

Масса объекта (в данном случае — жидкости) рассчитывается как:

Где:

• m — масса объекта (кг).

• V — объем объекта (м³).

• — плотность объекта (кг/м³).

**4. Испарение жидкости**

При нагреве жидкости до точки кипения она начинает испаряться. Количество тепла, необходимое для испарения некоторого объема жидкости, рассчитывается по формуле:

Где:

• — количество тепла, необходимое для испарения объема жидкости (Дж).

• L — удельная теплота испарения (Дж/кг).

• m — масса испаряющейся жидкости (кг).

Чтобы определить объем испарившейся жидкости, используется формула:

Где:

• — объем испарившейся жидкости (м³).

• — количество тепла, поглощенное жидкостью (Дж).

• L — удельная теплота испарения (Дж/кг).

**5. Теплоемкость жидкости**

Теплоемкость жидкости рассчитывается с использованием удельной теплоемкости (С) и массы жидкости:

Где:

• — общая теплоемкость жидкости (Дж/°C).

• C — удельная теплоемкость жидкости (Дж/кг°C).

• m — масса жидкости (кг).

Это значение используется для расчета того, насколько сильно изменится температура жидкости при поглощении или отдаче тепла.

**6. Общая формула расчета температуры жидкости**

При нагреве жидкости из окружающей среды температура жидкости обновляется по следующей формуле:

Где:

• — новая температура жидкости после теплового обмена (°C).

• — текущая температура жидкости (°C).

• — температура окружающей среды (°C).

• k — коэффициент теплопередачи (Вт/м²°C).

• S — площадь теплопередачи (м²).

• — общая теплоемкость жидкости (Дж/°C).

**7. Изменение температуры окружающей среды**

Тепловой обмен между жидкостью и окружающей средой также приводит к изменению температуры окружающей среды. Температура окружающей среды изменяется на величину, зависящую от общего теплового баланса между средой и объектом:

Где:

• — изменение температуры окружающей среды (°C).

• — общее количество тепла, переданное жидкостью или другими объектами в окружающую среду (Дж).

• — теплоемкость окружающей среды (Дж/°C).

**8. Уравнение теплового баланса**

Для объектов в системе происходит непрерывный тепловой обмен, который подчиняется закону сохранения энергии:

# 

# Интерфейс:

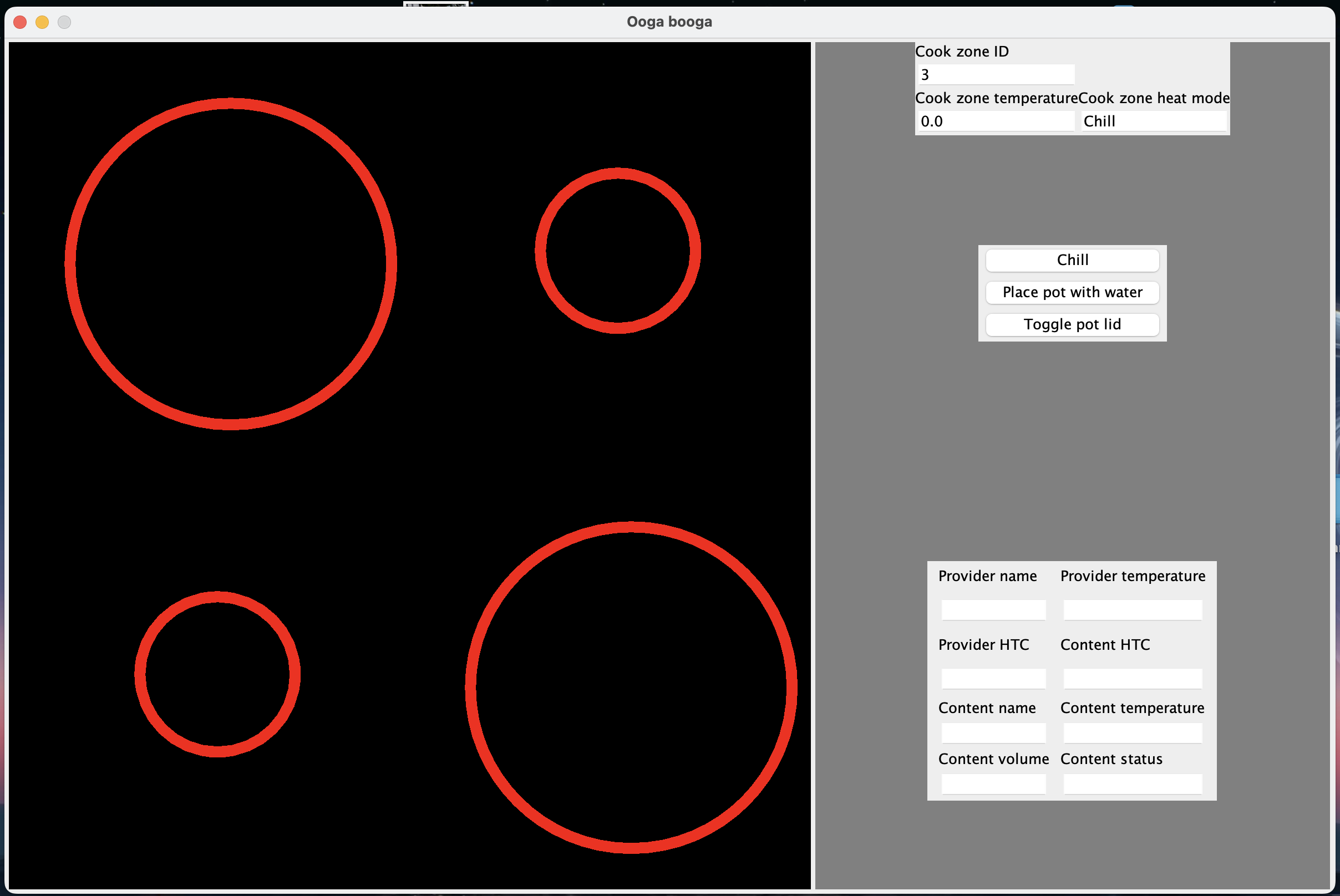


Рисунок 2. Интерфейс приложения

При клике на конфорку, можно выбрать режим нагрева, поставить/убрать кастрюлю, положить/убрать крышку с кастрюли

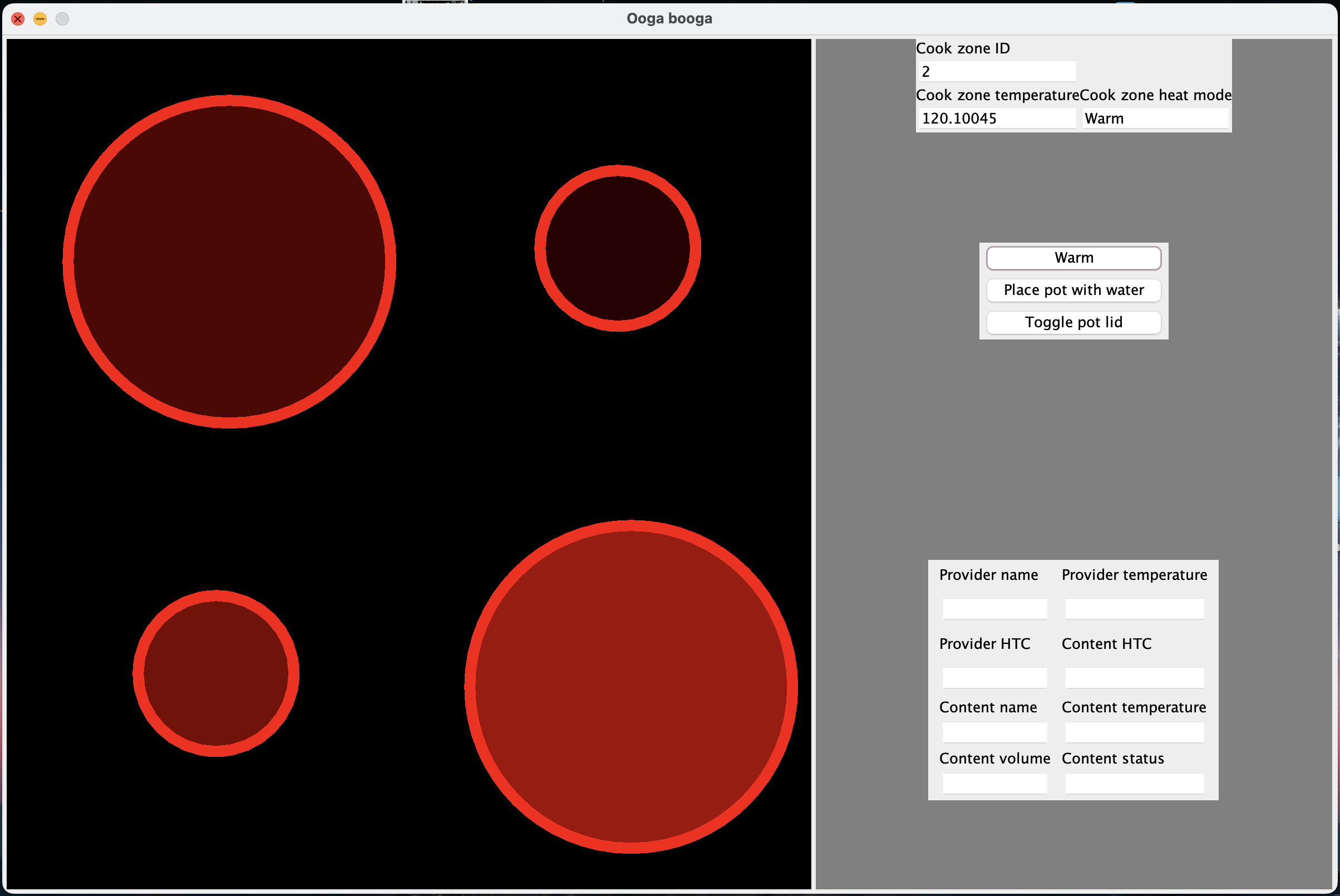


Рисунок 3. Функционирование конфорок плиты

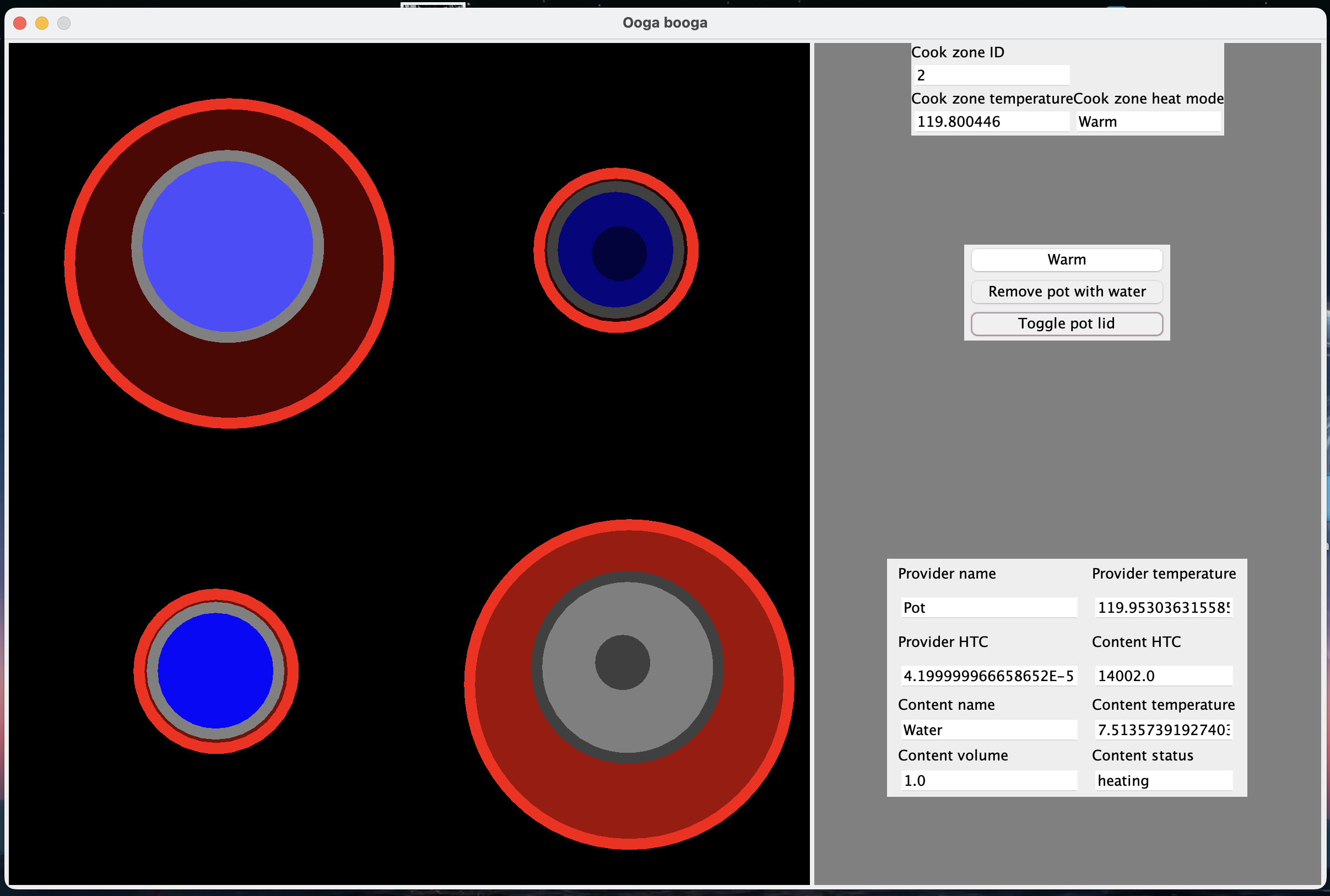


Рисунок 4. Отображение кастрюли на плите (с крышкой и без)

Информационная панель справа разделена на две части – сверху отображена информация о конфорке (температура, id, режим нагрева). Снизу отображена информация о том, что находится на конфорке.

# Устройство приложения изнутри:

1. *Как объект сообщает тепло другому объекту:*

Логика передачи тепла находится в HeatProvider. Эта модель поведения определяет функционал расчета получаемого тепла от другого HeatProvider, а также расчитывает температуру, которая будет сообщена дочернему объекту.



Рисунок 5. Метод включения конфорки плиты.

Поток передачи тепла объекту на конфорке

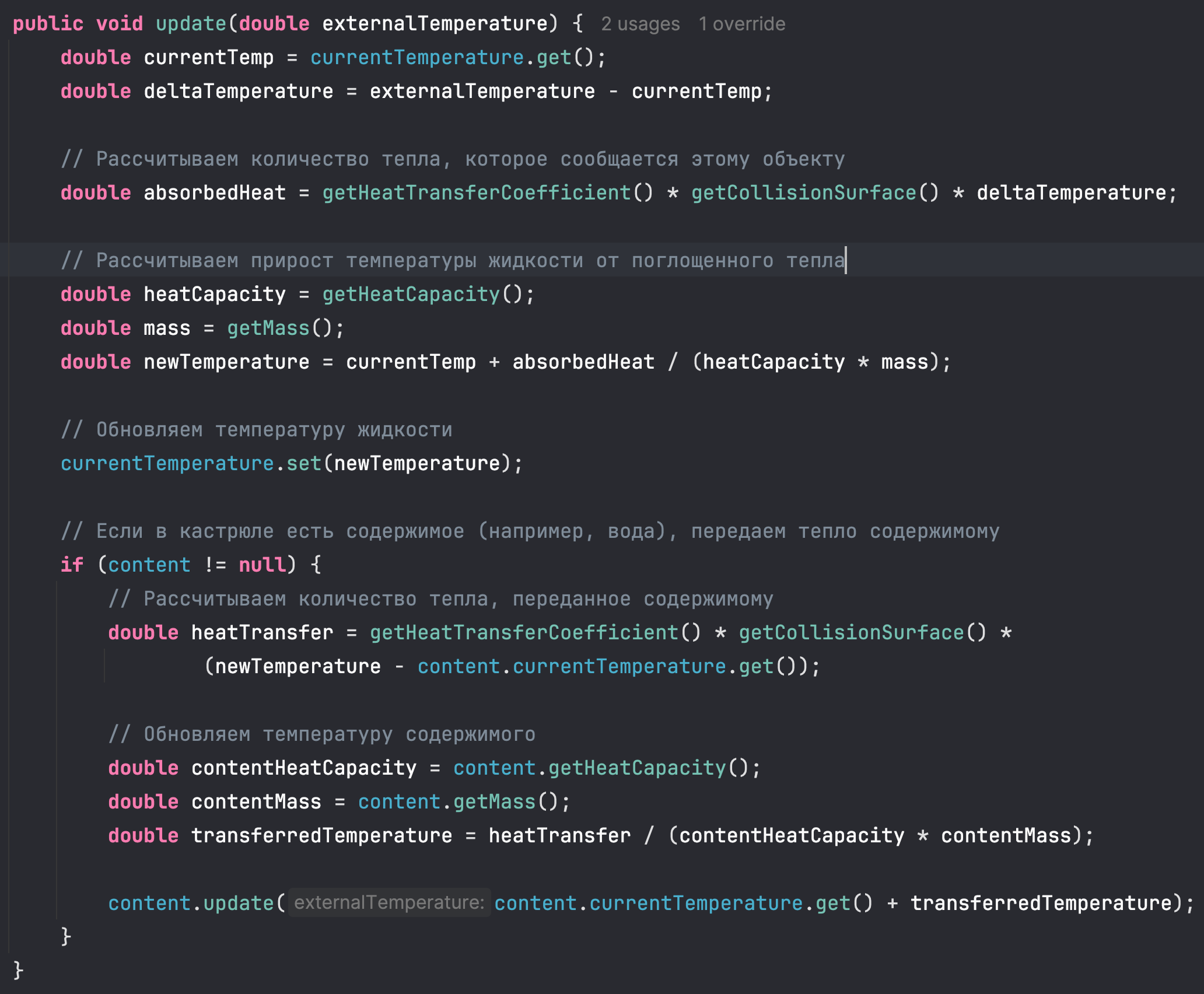
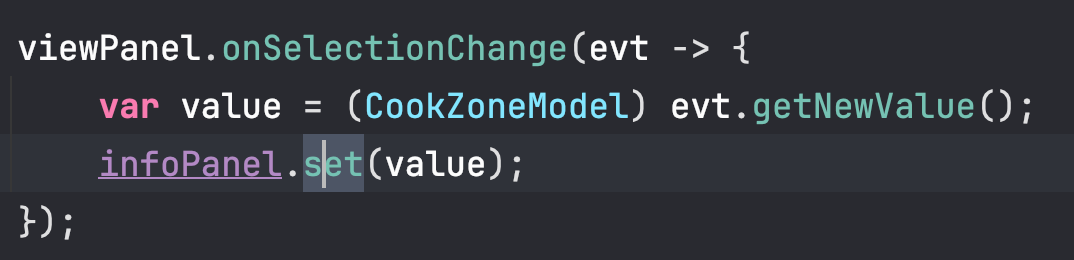


Рисунок 6. Метод для передачи температуры объекту HeatProvider

1. *Как приложение реагирует на действия пользователя:*

Взаимодействие пользователя с системой находится в CookZoneControlPanel. Все подписки на действия находятся там же.  
Чтобы не делать множество уровней вложенности (например, при необходимости среагировать на действие внутри другой подписки), был введен класс SignalBus и SignalListener. Мы подписываемся в MainFrame на клик по конфорке, и говорим слушаетелям, как им нужно отреагировать – можно назвать это правилом поведения.



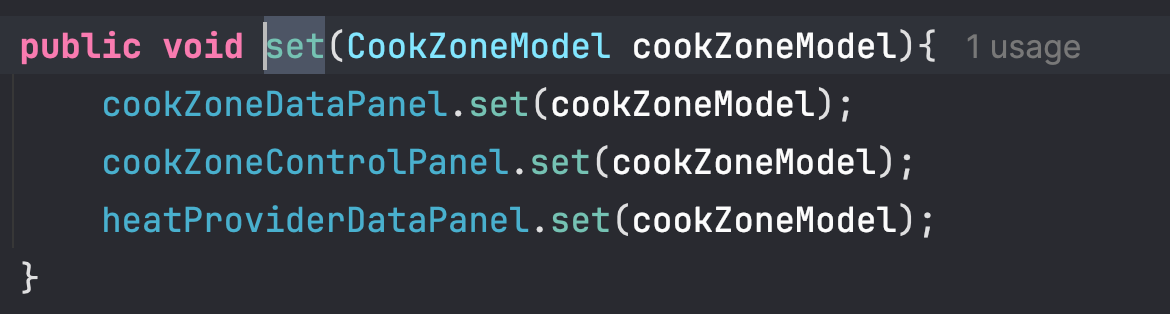


Рисунок 7. Реакция на смену выбранной конфорки в InfoPanel

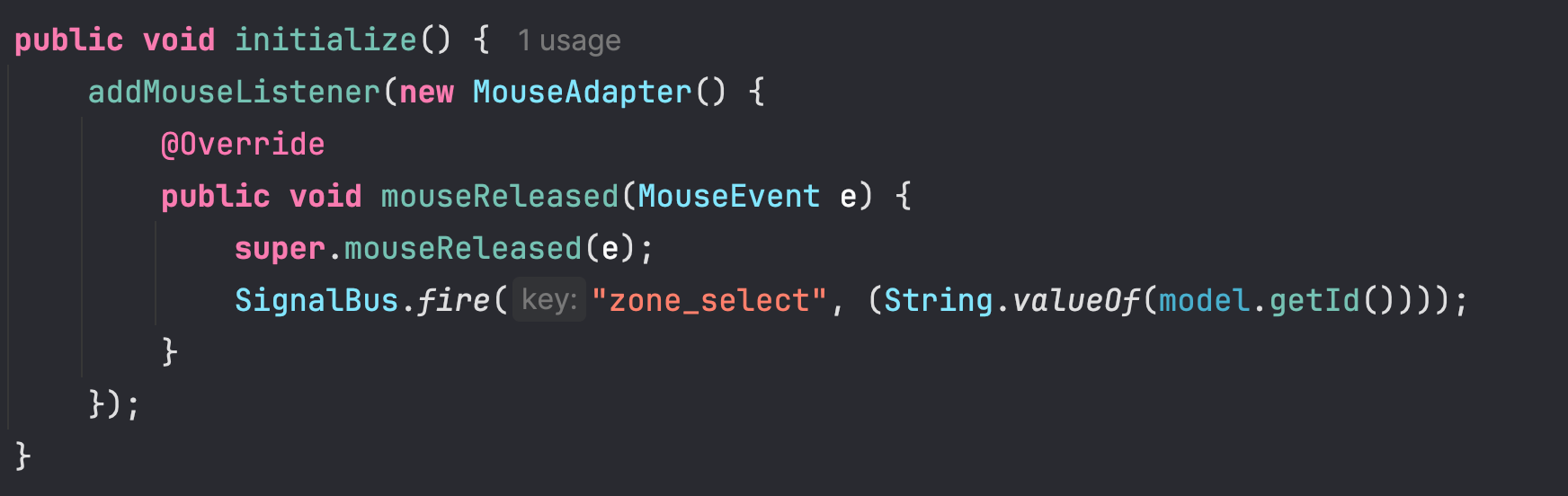


Рисунок 8. Выбор конфорки внутри CookZoneView



Рисунок 9. CooktopViewPanel - создание отображений конфорок и

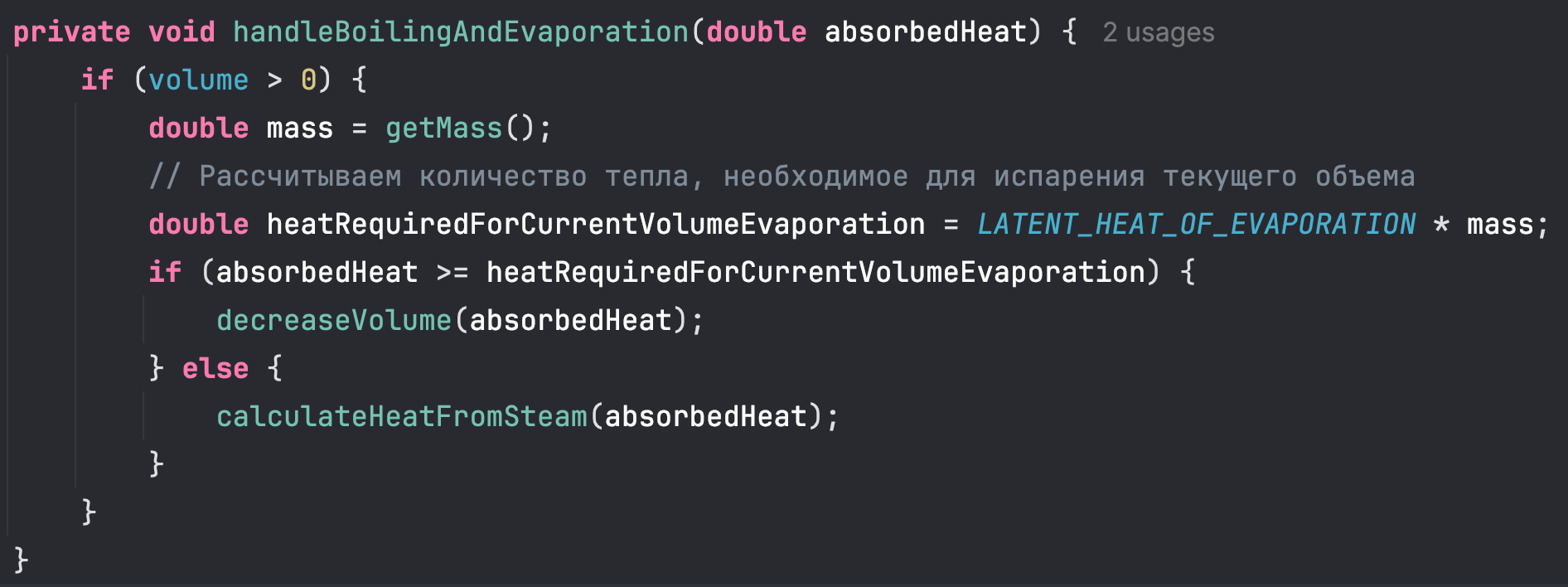
подписка на смену выбранной конфорки

1. *Как происходит переход между состояниями нагрева жидкости:*

⁠ Для представления такой логики, был введен класс StableHeatProvider. Если появится необходимость греть не только воду, можно будет просто поменять аргументы при создании экземпляра класса.



Рисунок 10. Метод для сообщения температуры жидкости. Расчет перехода между фазами нагрева



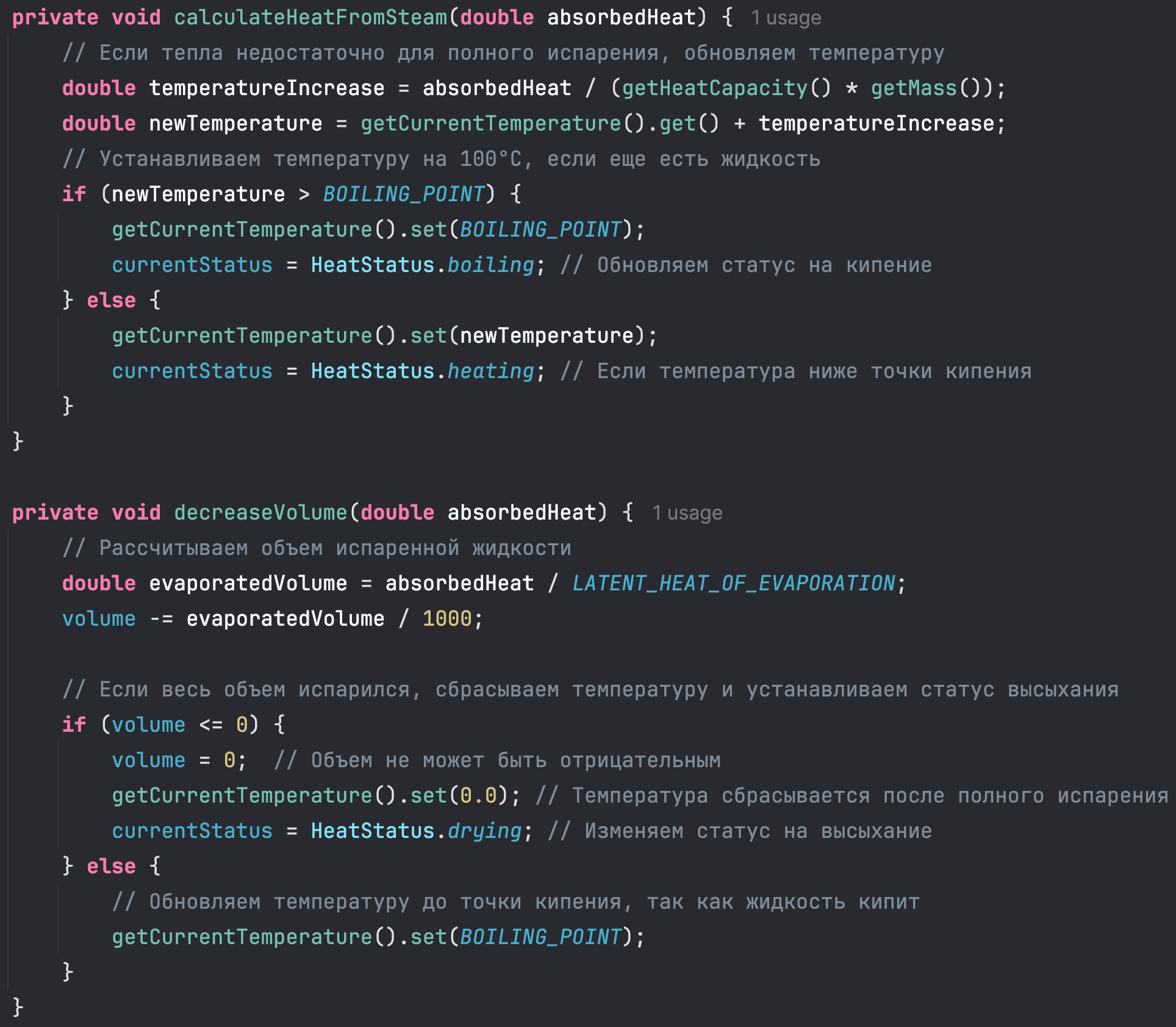


Рисунок 11. Методы для расчета испарения жидкости

# Известные проблемы устройства приложения:

1. Много логики в отображении. Можно было грамотнее завязать подписки, тем самым сняв лишнюю ответственность с отображений
2. Применение «магических цифр». Для упрощения тестирования и проверки работы, были использованы определенные константы в физических расчетах. Жертва точности симуляции в пользу времени разработки

# Вывод:

Следуя требованиям к заданию, была реализована симуляция варочной панели, которая включает в себя реагирование на действия пользователя и логику расчета передачи тепла между объектами