



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий

Кафедра вычислительной техники

КУРСОВАЯ РАБОТА

По дисциплине

«Объектно-ориентированное программирование»

(наименование дисциплины)

Тема курсовой работы

К_5 Моделирование работы кофе машины

(наименование темы)

Студент группы ИКБО-15-23

(учебная группа)

Ву Дык Зуй

(Фамилия Имя Отчество)

Зуй
(подпись студента)

Руководитель курсовой работы

доцент Пугуридзе З.И.

(Должность, звание, ученая степень)

З.И. Пугуридзе
(подпись руководителя)

Консультант

ст.преп. Данилович Е.С.

(Должность, звание, ученая степень)

Е.С. Данилович
(подпись консультанта)

Работа представлена к защите

«31» мая 2024 г.

Допущен к защите «31» мая

2024 г.

5 (отл)
7.06.2024

Москва 2024 г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий
Кафедра вычислительной техники

Утверждаю

Заведующий кафедрой

Платонова О.В.

ФИО

«21» февраля 2024г.

ЗАДАНИЕ

На выполнение курсовой работы

по дисциплине «Объектно-ориентированное программирование»

Студент Ву Дык Зуй

Группа ИКБО-15-23

Тема К_5 Моделирование работы кофе машины

Исходные данные:

1. Описания исходной иерархии дерева объектов.
2. Описание схемы взаимодействия объектов.
3. Множество команд для управления функционированием моделируемой системы.

Перечень вопросов, подлежащих разработке, и обязательного графического материала:

1. Построение версий программ.
2. Построение и работа с деревом иерархии объектов.
3. Взаимодействия объектов посредством интерфейса сигналов и обработчиков.
4. Блок-схемы алгоритмов.
5. Управление функционированием моделируемой системы

Срок представления к защите курсовой работы: до «31» мая 2024 г.

Задание на курсовую работу выдал

(Путуридзе З.Ш.)

ФИО консультанта

«21» февраля 2024 г.

Задание на курсовую работу получил

(Ву Д.З.)

ФИО исполнителя

«21» февраля 2024 г.

Москва 2024 г.

ОТЗЫВ

на курсовую работу

по дисциплине «Объектно-ориентированное программирование»

Студент Ву Дык Зуй группа ИКБО-15-23
(ФИО студента) (Группа)

Характеристика курсовой работы

Критерий	Да	Нет	Не полностью
1. Соответствие содержания курсовой работы указанной теме	+		
2. Соответствие курсовой работы заданию	+		
3. Соответствие рекомендациям по оформлению текста, таблиц, рисунков и пр.	+		
4. Полнота выполнения всех пунктов задания	+		
5. Логичность и системность содержания курсовой работы	+		
6. Отсутствие фактических грубых ошибок	+		

Замечаний:

Рекомендуемая оценка:

отлично



(Подпись)

ст. преподаватель Данилович Е.С.

(ФИО руководителя/консультанта)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 ПОСТАНОВКА ЗАДАЧИ.....	8
1.1 Описание входных данных.....	11
1.2 Описание выходных данных.....	12
2 МЕТОД РЕШЕНИЯ.....	14
3 ОПИСАНИЕ АЛГОРИТМОВ.....	19
3.1 Алгоритм метода phat_tin_hieu класса Suni.....	19
3.2 Алгоритм конструктора класса Suni.....	20
3.3 Алгоритм метода chuyen_tin_hieu класса Suni2.....	21
3.4 Алгоритм метода chuyen_tin_hieu1 класса Suni2.....	22
3.5 Алгоритм метода nhan_tin_hieu класса Suni2.....	24
3.6 Алгоритм метода chuyen_tin_hieu класса Suni3.....	26
3.7 Алгоритм метода nhan_tin_hieu класса Suni3.....	26
3.8 Алгоритм метода chuyen_tin_hieu класса Suni4.....	28
3.9 Алгоритм метода nhan_tin_hieu класса Suni4.....	28
3.10 Алгоритм метода chuyen_tin_hieu класса Suni5.....	29
3.11 Алгоритм метода nhan_tin_hieu класса Suni5.....	30
3.12 Алгоритм метода chuyen_tin_hieu класса Suni6.....	30
3.13 Алгоритм метода nhan_tin_hieu класса Suni6.....	31
3.14 Алгоритм метода chuyen_tin_hieu класса Suni7.....	31
3.15 Алгоритм метода nhan_tin_hieu класса Suni7.....	32
3.16 Алгоритм конструктора класса Suni.....	33
3.17 Алгоритм функции main.....	33
3.18 Алгоритм метода Taoj_say_phan_cap класса Suni_app.....	33
3.19 Алгоритм метода pearnet_tong класса Suni_app.....	36
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	38

5 КОД ПРОГРАММЫ.....	68
5.1 Файл main.cpp.....	68
5.2 Файл Suni2.cpp.....	68
5.3 Файл Suni2.h.....	70
5.4 Файл Suni3.cpp.....	70
5.5 Файл Suni3.h.....	71
5.6 Файл Suni4.cpp.....	72
5.7 Файл Suni4.h.....	73
5.8 Файл Suni5.cpp.....	73
5.9 Файл Suni5.h.....	74
5.10 Файл Suni6.cpp.....	74
5.11 Файл Suni6.h.....	75
5.12 Файл Suni7.cpp.....	75
5.13 Файл Suni7.h.....	76
5.14 Файл Suni_app.cpp.....	77
5.15 Файл Suni_app.h.....	78
5.16 Файл Suni.cpp.....	79
5.17 Файл Suni.h.....	85
6 ТЕСТИРОВАНИЕ.....	88
ЗАКЛЮЧЕНИЕ.....	93
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	94

ВВЕДЕНИЕ

Настоящая курсовая работа выполнена в соответствии с требованиями ГОСТ Единой системы программной документации (ЕСПД) [1]. Все этапы решения задач курсовой работы фиксированы, соответствуют требованиям, приведенным в методическом пособии для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [2-3] и методике разработки объектно-ориентированных программ [4-6].

Объектно-ориентированное программирование (ООП) - это модель программирования, основанная на концепции «объектов», в которой объекты содержат данные в полях, обычно называемых свойствами, а исходный код организован в методы. ООП может быть реализовано не только на C++, но и на других языках, таких как Python и Java. Не существует единого языка программирования, лучше всего подходящего для ООП, поскольку выбор языка зависит от конкретных потребностей проекта и навыков программиста. При использовании C++ для написания ООП этот язык имеет следующие преимущества:

- C++ обеспечивает высокую производительность и детальное управление памятью, что делает его хорошо подходящим для высокооптимизированных приложений.
- ООП тесно связано с C++, поскольку C++ спроектирован как объектно-ориентированный язык программирования, расширенный из языка C.
- C++ полностью поддерживает функции ООП, такие как классы, объекты, наследование, полиморфизм и инкапсуляция, позволяя программистам создавать сложные программы, используя концепции ООП.
- C++ - мощный и гибкий язык, который упрощает управление исходным

кодом и разработку программного обеспечения.

В объектно-ориентированном программировании **объект** — это конкретный экземпляр класса, который содержит как **свойства** (данные), так и **методы** (функции) для работы с этими данными. Объект — это конкретная реализация абстрактных концепций, определенных в классе.

- **Свойства** - это данные или информация, хранящаяся в объекте. Свойства обычно представляют собой переменные в классе и используются для описания состояния объекта.
- **Методы** - это функции или действия, которые объект может выполнять. Методы определены в классе и позволяют взаимодействовать со свойствами объекта или выполнять действия, связанные с объектом.

Конструктор и **деструктор** - это два специальных метода в объектно-ориентированном программировании, которые используются для управления жизненным циклом.

- **Конструктор** используется для инициализации объекта. Он вызывается автоматически при создании нового объекта. Он имеет то же имя, что и класс, которому он принадлежит.
- **Деструктор** используется для освобождения ресурсов, занятых объектом, когда он больше не нужен. Он вызывается автоматически при уничтожении объекта. Он имеет то же имя, что и класс, но с добавлением символа '~' в начале.

Указатели играют важную роль в объектно-ориентированном программировании, поскольку они обеспечивают гибкость и мощь для управления памятью и работы с объектами. Указатели становятся мощным инструментом в ООП, помогая программистам создавать мощные, гибкие и легко поддерживаемые приложения.

1 ПОСТАНОВКА ЗАДАЧИ

Надо моделировать работу кофемашины следующей конструкции. Кофе машина состоит из следующих элементов:

- пульт управления;
- устройство приема денег;
- устройство выдачи кофе;
- устройство возврата сдачи;
- экран отображения состояния и информации.

Пульт управления содержит кнопки:

- выбора кофе (множество кнопок);
- возврата денег.

Правила работы с кофе машиной.

Кофе машина готовится к работе следующим образом:

1. Задается количество сортов кофе (количество кнопок для выбора кофе) их названия и их стоимость, кратное 5 рублям. Загружается кофе. Подразумевается, что объем достаточен для работы.
2. Загружается заданное количество монет для выдачи сдачи с достоинством пять и десять рублей.
3. После этого выводиться сообщение о готовности кофе машины к работе.

После готовности кофе машины выполняются действия:

- ввод денег достоинством 5, 10, 50 или 100 рублей. При вводе денег осуществляется суммирование;
- выборе кофе, если денег достаточно, то выдается кофе и сдача, при наличии. Выводиться сообщение о готовности кофе машины к работе;
- выборе кофе, если денег недостаточно, сообщает о недостаточности средств;

- возврат денег, возвращаются все внесенные средства и выводиться сообщение о готовности кофе машины к работе.

Устройство возврата сдачи может вернуть только монеты с достоинством 5 и 10 рублей.

После ввода купюр достоинством 50 или 100 рублей проверяется возможность возврата внесенной суммы. Если монет с достоинством 5 и 10 рублей недостаточно, то купюры 50 или 100 не принимаются.

Возврат денег или сдача выдается максимальным количеством монет достоинством 10 рублей.

Нажатие на кнопки пульта управления и подача денег моделируется посредством клавиатурного ввода. Ввод делится на команды:

- «натуральное число кратное 5» - ввод денег;
- Coffee «наименование кофе» - нажатие кнопки сорта кофе (выбора кофе);
- Refund money – нажатие кнопки «вернуть деньги»;
- Cancel – завершение работы системы.

Отображение текста состояния кофе машины и результата операции моделируется посредством вывода на консоли.

Построить систему, которая использует объекты:

1. Объект «система».
2. Объект для чтения команд и данных. Считывает данные для подготовки и настройки кофе машины. После чтения очередной порции данных для настройки или данных команды, объект выдает сигнал с текстом полученных данных. Все данные настройки и данные команды синтаксический корректны.
3. Объект пульта управления, для отработки нажатия кнопок выбора кофе. Объект после нажатия кнопки анализирует достаточность средств и выдает соответствующий сигнал.

4. Объект, моделирующий устройства приема денег. После принятия очередной купюры производит суммирование и выдает сигнал, содержащий сумму введенных денег для отображения на экран.
5. Объект, моделирующий устройства возврата денег. Выдает сигнал, содержащий количество возвращаемой суммы. После выводится сообщение о готовности кофе машины к работе.
6. Объект, моделирующий устройства выдачи кофе. Выдает сигнал, содержащий текст . После выдачи кофе выдает сигнал о готовности кофе машины к работе.
7. Объект для вывода состояния или результата операции кофе машины на консоль.

Написать программу, реализующую следующий алгоритм:

1. Вызов метода объекта «система» `build_tree_objects ()`.
 - 1.1. Построение дерева иерархии объектов.
 - 1.2. Установка связей сигналов и обработчиков между объектами.
2. Вызов метода объекта «система» `exes_app ()`.
 - 2.1. Приведение всех объектов в состояние готовности.
 - 2.2. Цикл для обработки вводимых данных для настройки и команд.
 - 2.2.1. Выдача сигнала объекту для ввода команды.
 - 2.2.2. Отработка команды.
 - 2.3. После ввода команды «Cancel» завершить работу.

Все приведенные сигналы и соответствующие обработчики должны быть реализованы.

Все сообщения на консоль выводятся с новой строки.

В набор поддерживаемых команд добавить команду «SHOWTREE» и по этой команде вывести дерево иерархии объектов системы с отметкой о готовности и завершить работу программы.

1.1 Описание входных данных

Первая строка.

«натуральное число» «название кофе 1» ... «название кофе n»

Задаёт количество сортов кофе и их наименования в количестве не более 5. Выполняется операция загрузки кофе.

Вторая строка содержит целые числа кратные 5 в количестве сортов кофе. Каждое значение соответствует цене сорта кофе согласно индексу (порядку ввода). Выполняется операция настройки цен.

«натуральное число» «натуральное число» ...

Третья строка содержит исходное количество монет для выдачи сдачи. Выполняется операция первоначальной загрузки монет для сдачи.

«натуральное число» «натуральное число»

Первое число - количество пяти рублевых монет, второе число - количество десяти рублевых монет.

Последующие строки содержат команды (нажатия на кнопки или подача денег).

Подача денег, число 5, 10, 50 или 100.

«натуральное число»

Возврат денег.

Refund money

Выбор кофе.

Coffee «наименование кофе»

Последняя команда присутствует всегда.

Cancel

Пример ввода:

```
3 Espresso Americano Cappuchino
25 50 50
3 5
50
Coffee Cappuchino
10
10
10
Coffee Espresso
5
5
Refund money
5
100
Cancel
```

1.2 Описание выходных данных

Шаблоны текстов, которые отображаются на консоли:

Готов к работе, отображения в начале работы системы, после завершения загрузки кофе машины. Также отображается после завершения очередной операции и готовности кофе машины для обслуживания нового клиента.

Ready to work

Сумма после ввода очередной монеты или купюры.

The amount: «сумма денег»

Сообщение о готовности кофе:

Take the coffee «наименование кофе»

Сообщение для получения сдачи:

Take the change: 10 * «количество десяти рублевых монет» rub., 5 *
«количество пяти рублевых монет» rub.
Сообщение о недостаточности средств
There is not enough money

Сообщение для получения введенных денег обратно:

Take the money: 10 * «количество десяти рублевых монет» rub., 5 *
«количество пяти рублевых монет» rub.

Сообщение о возврате 50 или 100 рублевой купюры:

Take the money back, no change

Сообщение о завершении работы кофе машины:

Turned off

Пример вывода:

Ready to work
The amount: 50
Take the coffee Cappuchino
Ready to work
The amount: 10
The amount: 20
The amount: 30
Take the coffee Espresso
Take the change: 10 * 0 rub., 5 * 1 rub.
Ready to work
The amount: 5
The amount: 10
Take the money: 10 * 1 rub., 5 * 0 rub.
Ready to work
The amount: 5
Take the money back, no change
Turned off

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект cin класса istream предназначен для ввода с клавиатуры;
- объект cout класса ostream предназначен для вывода на экран;
- объект shl_thucthi класса Suni_app предназначен для создания дерева иерархии и запуска системы;
- объект указателя shl_tren класса Suni предназначен для определения головного объекта для нового объекта;
- объект указателя shl_duoі класса Suni предназначен для создания нового объекта в дереве иерархии;
- объект "Command Reader" класса Suni2 предназначен для чтения команд и данных;
- объект "Control Panel" класса Suni3 предназначен для отработки нажатия кнопок выбора кофе;
- объект "Money Receiver" класса Suni4 предназначен для устройства приема денег;
- объект "Change Returner" класса Suni5 предназначен для устройства возврата денег;
- объект "Display" класса Suni6 предназначен для вывода состояния или результата операции кофе машины на консоль;
- объект "Coffee Dispenser" класса Suni7 предназначен для устройства выдачи кофе;
- while - оператор цикла с предусловием;
- if...else - условный оператор;
- for - оператор цикла со счётчиком;
- new - оператор, используемый для динамического выделения памяти для

создания объекта или массива объектов;

- delete - оператор, используемый для освобождения памяти, динамически выделенной оператором "new";
- switch - оператор множественного выбора;
- do...while - оператор цикла с постусловием.

Класс Suni:

- свойства/поля:
 - о поле строковое для содержания имени объекта:
 - наименование — S_chuoioix;
 - тип — string;
 - модификатор доступа — protected;
 - о поле указатель на головной объект:
 - наименование — Suni_main;
 - тип — Suni*;
 - модификатор доступа — protected;
 - о поле вектор для содержания адресов подчинённых объектов:
 - наименование — Suni_luutru;
 - тип — vector <Suni*>;
 - модификатор доступа — protected;
 - о поле готовность объекта:
 - наименование — trangthai;
 - тип — int;
 - модификатор доступа — protected;
 - о поле вектор для хранения установленных связей:
 - наименование — Suni_ddeens;
 - тип — vector <ketnoi*>;
 - модификатор доступа — protected;

- о поле вектор для хранения имён кофе:
 - наименование — Ncoffee;
 - тип — vector <string>;
 - модификатор доступа — protected;
 - о поле вектор для хранения цен кофе:
 - наименование — Pcoffee;
 - тип — vector <int>;
 - модификатор доступа — protected;
 - о поле суммарные деньги в кофемашине:
 - наименование — sum;
 - тип — int;
 - модификатор доступа — protected;
 - о поле количество пяти рублевых монет:
 - наименование — c5;
 - тип — int;
 - модификатор доступа — protected;
 - о поле количество десяти рублевых монет:
 - наименование — c10;
 - тип — int;
 - модификатор доступа — protected;
 - о поле указатель на виртуальный головной объект корневого объекта:
 - наименование — Virt_obj;
 - тип — static Suni*;
 - модификатор доступа — protected;
- функционал:
 - о метод phat_tin_hieu — метод выдачи сигнала;
 - о метод Suni — инициализирование нового объекта класса Suni по

умолчанию;

- о метод `Sun1` — инициализирование нового объекта класса `Sun1` с параметрами.

Класс `Sun2`:

- функционал:
 - о метод `chuyen_tin_hieu` — метод передачи сигнала класса `Sun2`;
 - о метод `chuyen_tin_hieu1` — метод установки данных в кофемашине;
 - о метод `nhan_tin_hieu` — метод получения сигнала класса `Sun2`.

Класс `Sun3`:

- функционал:
 - о метод `chuyen_tin_hieu` — метод передачи сигнала класса `Sun3`;
 - о метод `nhan_tin_hieu` — метод получения сигнала класса `Sun3`.

Класс `Sun4`:

- функционал:
 - о метод `chuyen_tin_hieu` — метод передачи сигнала класса `Sun4`;
 - о метод `nhan_tin_hieu` — метод получения сигнала класса `Sun4`.

Класс `Sun5`:

- функционал:
 - о метод `chuyen_tin_hieu` — метод передачи сигнала класса `Sun5`;
 - о метод `nhan_tin_hieu` — метод получения сигнала класса `Sun5`.

Класс `Sun6`:

- функционал:
 - о метод `chuyen_tin_hieu` — метод передачи сигнала класса `Sun6`;
 - о метод `nhan_tin_hieu` — метод получения сигнала класса `Sun6`.

Класс `Sun7`:

- функционал:
 - о метод `chuyen_tin_hieu` — метод передачи сигнала класса `Sun7`;

- о метод nhan_tin_hieu — метод получения сигнала класса Suni7.

Класс Suni_app:

- функционал:
 - о метод Taoj_say_phan_cap — метод создания дерева иерархии;
 - о метод pearnet_tong — метод запуска системы.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	Suni				
		Suni2	public		2
		Suni3	public		3
		Suni4	public		4
		Suni5	public		5
		Suni6	public		6
		Suni7	public		7
		Suni_app	public		8
2	Suni2				
3	Suni3				
4	Suni4				
5	Suni5				
6	Suni6				
7	Suni7				
8	Suni_app				

3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

3.1 Алгоритм метода `phat_tin_hieu` класса `Sunì`

Функционал: метод выдачи сигнала.

Параметры: нет.

Возвращаемое значение: `TYPE_SIGNAL ptr_tin_hieu, string tin_nhan, Sunì*`
`S_ptr` - указатель на метод сигнала текущего объекта, строковой параметр для содержания сообщения и указатель на целевой объект.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `phat_tin_hieu` класса `Sunì`

№	Предикат	Действия	№ перехода
1		Объявление указателя <code>ptr_nhan_th</code> на метод обработчика целевого объекта для выдачи сигнала целевому объекту	2
2		Объявление указателя <code>ptr_Sunì</code> класса <code>Sunì</code> для содержания адреса целевого объекта	3
3	Значение свойства <code>trangthai</code> равно 0		∅
		Вызов метода указателя <code>ptr_tin_hieu</code> текущего объекта с параметром <code>tin_nhan</code>	4
4		Инициализирование целочисленной переменной <code>i</code> для начала цикла: <code>int i = 0</code>	5
5	Значение <code>i</code> меньше		6

№	Предикат	Действия	№ перехода
	размерности вектора Sun _i _ddeens текущего объекта		
			∅
6	(Значение параметра ptr _{tin_hieu} равно значению переменной ptr _{tin_hieu} структуры элемента вектора Sun _i _ddeens по индексу i) и (Значение свойства trangthai объекта указателя ptr_Sun _i структуры элемента вектора Sun _i _ddeens по индексу i не равно 0) и (Значение указателя ptr_Sun _i структуры элемента вектора Sun _i _ddeens по индексу i равно значению параметра S_ptr)	Присвоение значения переменной ptr _{nhan_th} структуры элемента вектора Sun _i _ddeens по индексу i переменной ptr _{nhan_th}	7
		Инкремент переменной i	5
7		Присвоение значения переменной ptr_Sun _i структуры элемента вектора Sun _i _ddeens по индексу i указателю ptr_Sun _i	8
8		Вызов метода указателя ptr _{nhan_th} объекта указателя ptr_Sun _i с параметром tin _{nhan} ; Инкремент переменной i	5

3.2 Алгоритм конструктора класса Sun_i

Функционал: инициализирование нового объекта класса Sun_i с параметрами.

Параметры: Suni* Suni_main, string S_chuooiх - указатель класса Suni на головной объект текущего объекта и строковой параметр для содержания наименования текущего объекта.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Suni

№	Предикат	Действия	№ перехода
1		Присвоение значения параметра S_chuooiх свойству S_chuooiх	2
2		Присвоение значения параметра Suni_main свойству Suni_main	3
3	Свойство Suni_main не равно nullptr	Добавление адреса текущего объекта вектору Suni_luutru головного объекта текущего объекта	Ø
		Добавление адреса текущего объекта вектору Suni_luutru объекта указателя Virt_obj	Ø

3.3 Алгоритм метода chuyen_tin_hieu класса Suni2

Функционал: метод передачи сигнала класса Suni2.

Параметры: string& tin_nhan - ссылочный строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода chuyen_tin_hieu класса Suni2

№	Предикат	Действия	№ перехода
1	(Значение параметра tin_nhan равно "money") или (значение параметра tin_nhan равно "change")	Объявления целочисленных переменных sx5 и sx10 для расчёта суммы, подлежащую возврату	2

№	Предикат	Действия	№ перехода
	Значение параметра tin_nhan равно "false"	Присвоение значения "\nTake the money back, no change" параметру tin_nhan	Ø
			Ø
2		Присвоение частного значения свойства sum корневого объекта и значения 10 переменной cx10	3
3		Уменьшение значения свойства sum корневого объекта на произведение значения переменной cx10 и значения 10	4
4		Присвоение частного значения свойства sum корневого объекта и значения 5 переменной cx5	5
5		Уменьшение значения свойства sum корневого объекта на произведение значения переменной cx5 и значения 5	6
6		Уменьшение значения свойства c5 корневого объекта на значение переменной cx5	7
7		Уменьшение значения свойства c10 корневого объекта на значение переменной cx10	8
8	Значение параметра tin_nhan равно "money"	Присвоение значения (строковое значение переменной cx10 + "/" + строковое значение переменной cx5) параметру tin_nhan	Ø
	Значение параметра tin_nhan равно "change"	Присвоение значения (строковое значение переменной cx10 + "\$" + строковое значение переменной cx5) параметру tin_nhan	Ø
			Ø

3.4 Алгоритм метода chuyen_tin_hieu1 класса Suni2

Функционал: метод установки данных в кофемашине.

Параметры: string& tin_nhan - ссылочный строковой параметр для

содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *chuyen_tin_hieu1* класса *Sun12*

№	Предикат	Действия	№ перехода
1		Объявления целочисленных переменных <i>n</i> и <i>p</i> для содержания кол-ва кофе и содержания цены кофе	2
2		Присвоение целочисленного значения параметра <i>tin_nhan</i> переменной <i>n</i>	3
3		Объявление строковой переменной <i>Sname</i> для содержания имени кофе	4
4		Инициализирование целочисленной переменной <i>i</i> для начала цикла: <i>int i = 0</i>	5
5	Значение переменной <i>i</i> меньше значения переменной <i>n</i>	Ввод значения переменной <i>Sname</i>	6
		Инициализирование целочисленной переменной <i>i</i> для начала цикла: <i>int i = 0</i>	7
6		Добавление значения переменной <i>Sname</i> вектору <i>Ncoffee</i> корневого объекта; Инкремент переменной <i>i</i>	5
7	Значение переменной <i>i</i> меньше значения переменной <i>n</i>	Ввод значения переменной <i>p</i>	8
		Ввод значения свойства <i>s5</i> корневого объекта	9
8		Добавление значения переменной <i>p</i> вектору <i>Pcoffee</i> корневого объекта; Инкремент переменной <i>i</i>	7
9		Ввод значения свойства <i>s10</i> корневого объекта	10
10		Присвоение значения "Ready to work" параметру	∅

№	Предикат	Действия	№ перехода
		tin_nhan	

3.5 Алгоритм метода nhan_tin_hieu класса Suni2

Функционал: метод получения сигнала класса Suni2.

Параметры: string tin_nhan - строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода nhan_tin_hieu класса Suni2

№	Предикат	Действия	№ перехода
1	Значение параметра tin_nhan равно "Refund"	Ввод значения параметра tin_nhan	2
	Значение параметра tin_nhan равно "Coffee"	Ввод значения параметра tin_nhan	3
	Значение параметра tin_nhan равно "Cancel"	Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, строковым значением "\nTurned off" и адресом объекта с именем "Display"	Ø
	Значение параметра tin_nhan равно "SHOWTREE"	Вывод: переход на новую строку	4
	(Целочисленное значение параметра tin_nhan равно 50) или (целочисленное значение параметра tin_nhan равно 100)		5
	(Целочисленное значение параметра tin_nhan равно 5)		6

№	Предикат	Действия	№ перехода
	или (целочисленное значение параметра tin_nhan равно 10)		
			Ø
2		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением tin_nhan и адресом объекта с именем "Change Returner"	Ø
3		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением tin_nhan и адресом объекта с именем "Control Panel"	Ø
4		Вызов метода pearnet1() корневого объекта для вывода дерева иерархии объектов системы с отметкой о готовности	Ø
5	Сумма произведения значения свойства с5 корневого объекта и значения 5 и произведения значения свойства с10 корневого объекта и значения 10 меньше целочисленного значения параметра tin_nhan	Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, строковым значением "false" и адресом объекта с именем "Display"	Ø
		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением tin_nhan и адресом объекта с именем "Money Receiver"	Ø
6		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2,	7

№	Предикат	Действия	№ перехода
		значением tin_nhan и адресом объекта с именем "Money Receiver"	
7	Значение параметра tin_nhan равно "5"	Инкремент значения свойства c5 корневого объекта	Ø
	Значение параметра tin_nhan равно "10"	Инкремент значения свойства c10 корневого объекта	Ø
			Ø

3.6 Алгоритм метода chuyen_tin_hieu класса Suni3

Функционал: метод передачи сигнала класса Suni3.

Параметры: string& tin_nhan - ссылочный строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода chuyen_tin_hieu класса Suni3

№	Предикат	Действия	№ перехода
1	Значение параметра tin_nhan равно "false"	Присвоение значения "\There is not enough money" параметру tin_nhan	Ø
			Ø

3.7 Алгоритм метода nhan_tin_hieu класса Suni3

Функционал: метод получения сигнала класса Suni3.

Параметры: string tin_nhan - строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода *phan_tin_hieu* класса *SunI3*

№	Предикат	Действия	№ перехода
1		Инициализирование булевой переменной true	2
2		Инициализирование целочисленной переменной i для начала цикла: int i = 0	3
3	Значение переменной i меньше размерности вектора Ncoffee корневого объекта		4
			7
4	(Значение параметра tin_nhan равно значению элемента вектора Ncoffee корневого объекта по индексу i) и (значение элемента вектора Pcoffee корневого объекта по индексу i меньше или равно значению свойства sum корневого объекта)	Уменьшение значения свойства sum корневого объекта на значение элемента вектора Pcoffee корневого объекта по индексу i	5
		Инкремент значения переменной i	3
5		Присвоение значения false переменной check	6
6	Значение свойства sum корневого объекта равно 0	Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса SunI3, значением tin_nhan и адресом объекта с именем "Display"; Инкремент значения переменной i	3
		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса SunI3, значением (tin_nhan + "/change") и адресом	3

№	Предикат	Действия	№ перехода
		объекта с именем "Display"; Инкремент значения переменной i	
7	Значение переменной check равно true	Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni3, строковым значением "false" и адресом объекта с именем "Coffee Dispenser"	∅
			∅

3.8 Алгоритм метода chuyen_tin_hieu класса Suni4

Функционал: метод передачи сигнала класса Suni4.

Параметры: string& tin_nhan - ссылочный строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода chuyen_tin_hieu класса Suni4

№	Предикат	Действия	№ перехода
1		Присвоение значения ("\\nThe amount: " + значение параметра tin_nhan) параметру tin_nhan	∅

3.9 Алгоритм метода nhan_tin_hieu класса Suni4

Функционал: метод получения сигнала класса Suni4.

Параметры: string tin_nhan - строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода *nhan_tin_hieu* класса *Suni4*

№	Предикат	Действия	№ перехода
1		Увеличение значения свойства <i>sum</i> корневого объекта на целочисленное значение параметра <i>tin_nhan</i>	2
2		Вызов метода <i>phat_tin_hieu()</i> с параметрами адресом на метод <i>chuyen_tin_hieu()</i> класса <i>Suni4</i> , строковым значением свойства <i>sum</i> корневого объекта и адресом объекта с именем "Coffee Dispenser"	Ø

3.10 Алгоритм метода *chuyen_tin_hieu* класса *Suni5*

Функционал: метод передачи сигнала класса *Suni5*.

Параметры: *string& tin_nhan* - ссылочный строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа *void*).

Алгоритм метода представлен в таблице 11.

Таблица 11 – Алгоритм метода *chuyen_tin_hieu* класса *Suni5*

№	Предикат	Действия	№ перехода
1	"/" не найдено в значении параметра <i>tin_nhan</i>	Присвоение значения (" Take the change: 10 * " + значение подстроки параметра <i>tin_nhan</i> , который начинается с индекса 0 и длится (индекс символа "\$" в параметре <i>tin_nhan</i>) элементов, + " rub., 5 * " + значение подстроки параметра <i>tin_nhan</i> , который начинается с (индекс символа "\$" в параметре <i>tin_nhan</i> + 1), + " rub.") параметру <i>tin_nhan</i>	Ø
		Присвоение значения (" Take the money: 10 * " + значение подстроки параметра <i>tin_nhan</i> , который начинается с индекса 0 и длится (индекс символа "/" в параметре <i>tin_nhan</i>) элементов, + " rub., 5 * "	Ø

№	Предикат	Действия	№ перехода
		+ значение подстроки параметра tin_nhan, который начинается с (индекс символа "/" в параметре tin_nhan + 1), + " rub.\nReady to work") параметру tin_nhan	

3.11 Алгоритм метода nhan_tin_hieu класса Suni5

Функционал: метод получения сигнала класса Suni5.

Параметры: string tin_nhan - строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 12.

Таблица 12 – Алгоритм метода nhan_tin_hieu класса Suni5

№	Предикат	Действия	№ перехода
1		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni5, значением tin_nhan и адресом объекта с именем "Coffee Dispenser"	Ø

3.12 Алгоритм метода chuyen_tin_hieu класса Suni6

Функционал: метод передачи сигнала класса Suni6.

Параметры: string& tin_nhan - ссылочный строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 13.

Таблица 13 – Алгоритм метода *chuyen_tin_hieu* класса *Sunib*

№	Предикат	Действия	№ перехода
1			Ø

3.13 Алгоритм метода *nhan_tin_hieu* класса *Sunib*

Функционал: метод получения сигнала класса *Sunib*.

Параметры: *string tin_nhan* - строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа *void*).

Алгоритм метода представлен в таблице 14.

Таблица 14 – Алгоритм метода *nhan_tin_hieu* класса *Sunib*

№	Предикат	Действия	№ перехода
1		Вывод значения параметра <i>tin_nhan</i>	Ø

3.14 Алгоритм метода *chuyen_tin_hieu* класса *Sunib7*

Функционал: метод передачи сигнала класса *Sunib7*.

Параметры: *string& tin_nhan* - ссылочный строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа *void*).

Алгоритм метода представлен в таблице 15.

Таблица 15 – Алгоритм метода *chuyen_tin_hieu* класса *Sunib7*

№	Предикат	Действия	№ перехода
1	Значение параметра <i>tin_nhan</i> не равно "\nReady to work"	Присвоение значения (" <i>nTake the coffee " + значение параметра <i>tin_nhan</i>) параметру <i>tin_nhan</i></i>	Ø
			Ø

3.15 Алгоритм метода `nhan_tin_hieu` класса `Sun7`

Функционал: метод получения сигнала класса `Sun7`.

Параметры: `string tin_nhan` - строковой параметр для содержания сообщения в сигнале.

Возвращаемое значение: отсутствует (метод типа `void`).

Алгоритм метода представлен в таблице 16.

Таблица 16 – Алгоритм метода `nhan_tin_hieu` класса `Sun7`

№	Предикат	Действия	№ перехода
1	<code>"/"</code> не найдено в значении параметра <code>tin_nhan</code>	Вызов метода <code>phat_tin_hieu()</code> с параметрами адресом на метод <code>chuyen_tin_hieu()</code> класса <code>Sun7</code> , значением <code>tin_nhan</code> и адресом объекта с именем <code>"Coffee Dispenser"</code>	3
		Вызов метода <code>phat_tin_hieu()</code> с параметрами адресом на метод <code>chuyen_tin_hieu()</code> класса <code>Sun7</code> , значением подстроки параметра <code>tin_nhan</code> , который начинается с индекса 0 и длится (индекс символа <code>"/"</code> в параметре <code>tin_nhan</code>) элементов и адресом объекта с именем <code>"Coffee Dispenser"</code>	2
2		Вызов метода <code>phat_tin_hieu()</code> с параметрами адресом на метод <code>chuyen_tin_hieu()</code> класса <code>Sun7</code> , значением подстроки параметра <code>tin_nhan</code> , который начинается с (индекс символа <code>"/"</code> в параметре <code>tin_nhan</code> + 1) и и адресом объекта с именем <code>"Change Returner"</code>	3
3		Вызов метода <code>phat_tin_hieu()</code> с параметрами адресом на метод <code>chuyen_tin_hieu()</code> класса <code>Sun7</code> , строковым значением <code>"\nReady to work"</code> и адресом объекта с именем <code>"Coffee Dispenser"</code>	∅

3.16 Алгоритм конструктора класса Suni

Функционал: инициализирование нового объекта класса Suni по умолчанию.

Параметры: нет.

Алгоритм конструктора представлен в таблице 17.

Таблица 17 – Алгоритм конструктора класса Suni

№	Предикат	Действия	№ перехода
1			Ø

3.17 Алгоритм функции main

Функционал: точка входа в программу - основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 18.

Таблица 18 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Создание объекта shl_thucthi класса Suni_app с параметром nullptr для создания и запуска дерева иерархии объектов	2
2		Вызов метода Taoj_say_phan_cap() объекта shl_thucthi	3
3		Возврат возвр. зн. метода pearnet_tong() объекта shl_thucthi	Ø

3.18 Алгоритм метода Taoj_say_phan_cap класса Suni_app

Функционал: метод создания дерева иерархии.

Параметры: нет.

Возвращаемое значение: отсутствует (метод типа void).

Алгоритм метода представлен в таблице 19.

Таблица 19 – Алгоритм метода *Taoj_say_phan_car* класса *Suni_app*

№	Предикат	Действия	№ перехода
1		Создание объекта класса Suni2 с параметрами адресом текущего объекта и значением "Command Reader", и инициализирование указателя ptr_shl_ddeens2 класса Suni адресом нового объекта для создания объекта для чтения команд и данных	2
2		Создание объекта класса Suni3 с параметрами адресом текущего объекта и значением "Control Panel", и инициализирование указателя ptr_shl_ddeens3 класса Suni адресом нового объекта для создания объекта для отработки нажатия кнопок выбора кофе	3
3		Создание объекта класса Suni4 с параметрами адресом текущего объекта и значением "Money Receiver", и инициализирование указателя ptr_shl_ddeens4 класса Suni адресом нового объекта для создания устройства приема денег	4
4		Создание объекта класса Suni5 с параметрами адресом текущего объекта и значением "Change Returner", и инициализирование указателя ptr_shl_ddeens5 класса Suni адресом нового объекта для создания устройства возврата денег	5
5		Создание объекта класса Suni6 с параметрами адресом текущего объекта и значением "Coffee Dispenser", и инициализирование указателя ptr_shl_ddeens6 класса Suni адресом нового объекта для создания устройства выдачи кофе	6
6		Создание объекта класса Suni7 с параметрами адресом текущего объекта и значением "Display", и инициализирование указателя ptr_shl_ddeens7 класса Suni адресом нового объекта для вывода состояния или результата операции кофемашины на консоль	7
7		Вызов метода caidat_ketnoi() текущего объекта с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens2 и адресом на метод nhan_tin_hieu() класса Suni2	8

№	Предикат	Действия	№ перехода
8		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens2 с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens4 и адресом на метод nhan_tin_hieu() класса Suni4	9
9		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens4 с параметрами адресом на метод chuyen_tin_hieu() класса Suni4, значением ptr_shl_ddeens6 и адресом на метод nhan_tin_hieu() класса Suni6	10
10		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens2 с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens6 и адресом на метод nhan_tin_hieu() класса Suni6	11
11		Вызов метода caidat_ketnoi() текущего объекта с параметрами адресом на метод chuyen_tin_hieu1() класса Suni2, значением ptr_shl_ddeens6 и адресом на метод nhan_tin_hieu() класса Suni6	12
12		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens2 с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens5 и адресом на метод nhan_tin_hieu() класса Suni5	13
13		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens5 с параметрами адресом на метод chuyen_tin_hieu() класса Suni5, значением ptr_shl_ddeens6 и адресом на метод nhan_tin_hieu() класса Suni6	14
14		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens2 с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens3 и адресом на метод nhan_tin_hieu() класса Suni3	15
15		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens3 с параметрами адресом на метод chuyen_tin_hieu() класса Suni3, значением ptr_shl_ddeens6 и адресом на метод nhan_tin_hieu() класса	16

№	Предикат	Действия	№ перехода
		Suni6	
16		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens3 с параметрами адресом на метод chuyen_tin_hieu() класса Suni3, значением ptr_shl_ddeens7 и адресом на метод nhan_tin_hieu() класса Suni7	17
17		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens7 с параметрами адресом на метод chuyen_tin_hieu() класса Suni7, значением ptr_shl_ddeens6 и адресом на метод nhan_tin_hieu() класса Suni6	18
18		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens2 с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens5 и адресом на метод nhan_tin_hieu() класса Suni5	19
19		Вызов метода caidat_ketnoi() объекта указателя ptr_shl_ddeens7 с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, значением ptr_shl_ddeens5 и адресом на метод nhan_tin_hieu() класса Suni5	∅

3.19 Алгоритм метода pearnet_tong класса Suni_app

Функционал: метод запуска системы.

Параметры: нет.

Возвращаемое значение: целочисленное - код возврата функции main.

Алгоритм метода представлен в таблице 20.

Таблица 20 – Алгоритм метода pearnet_tong класса Suni_app

№	Предикат	Действия	№ перехода
1		Вызов метода Active() объекта указателя Virt_obj	2
2		Объявление строковой переменной command для	3

№	Предикат	Действия	№ перехода
		содержания команды	
3		Ввод значения command	4
4		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu1() класса Suni2, command и адресом объекта с именем "Coffee Dispenser"	5
5		Ввод значения переменной command	6
6		Вызов метода phat_tin_hieu() с параметрами адресом на метод chuyen_tin_hieu() класса Suni2, command и адресом объекта с именем "Command Reader"	7
7	(Значение переменной command не равно "Cancel") и (значение переменной command не равно "SHOWTREE")		5
			∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-30.

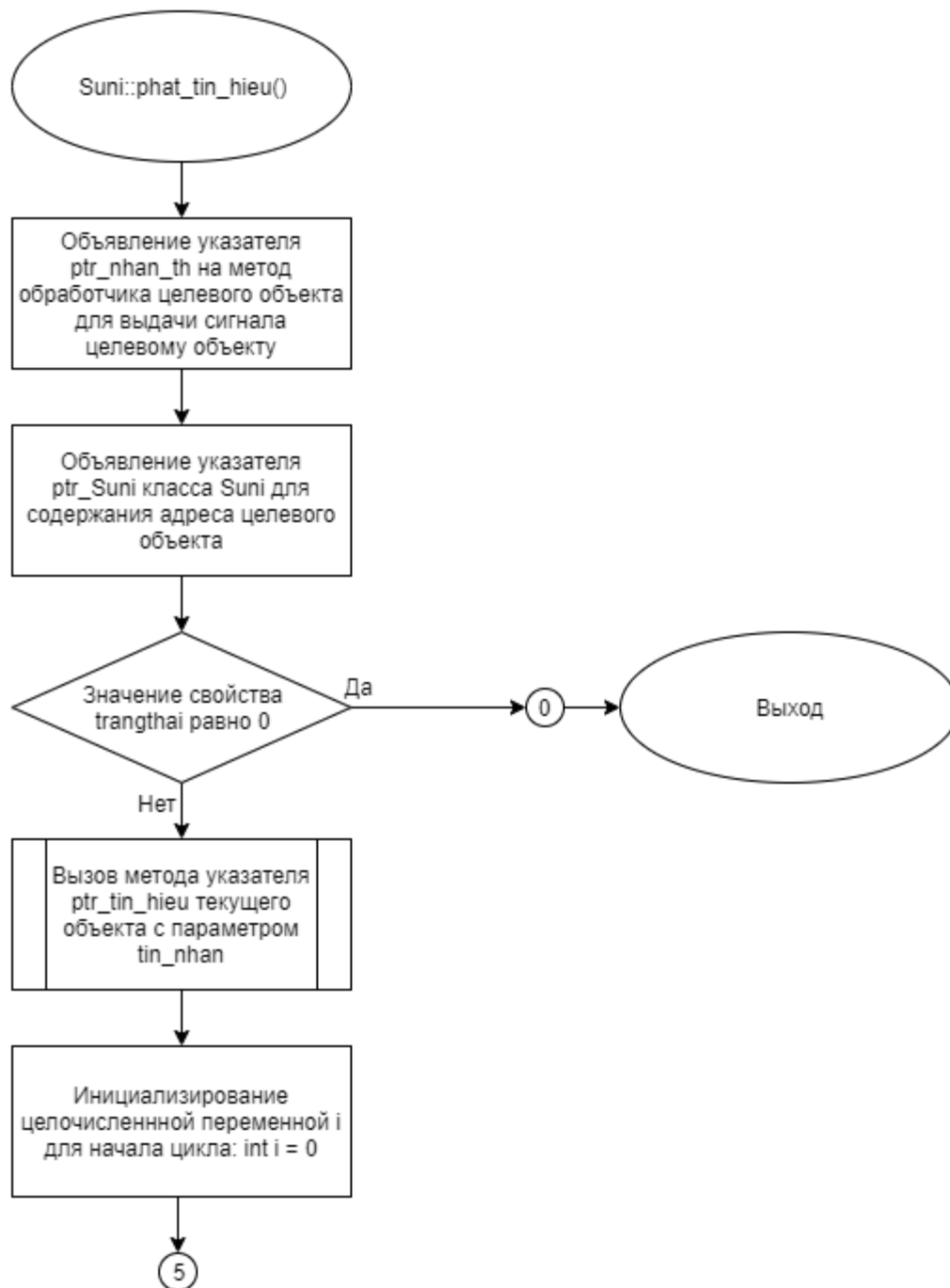


Рисунок 1 – Блок-схема алгоритма

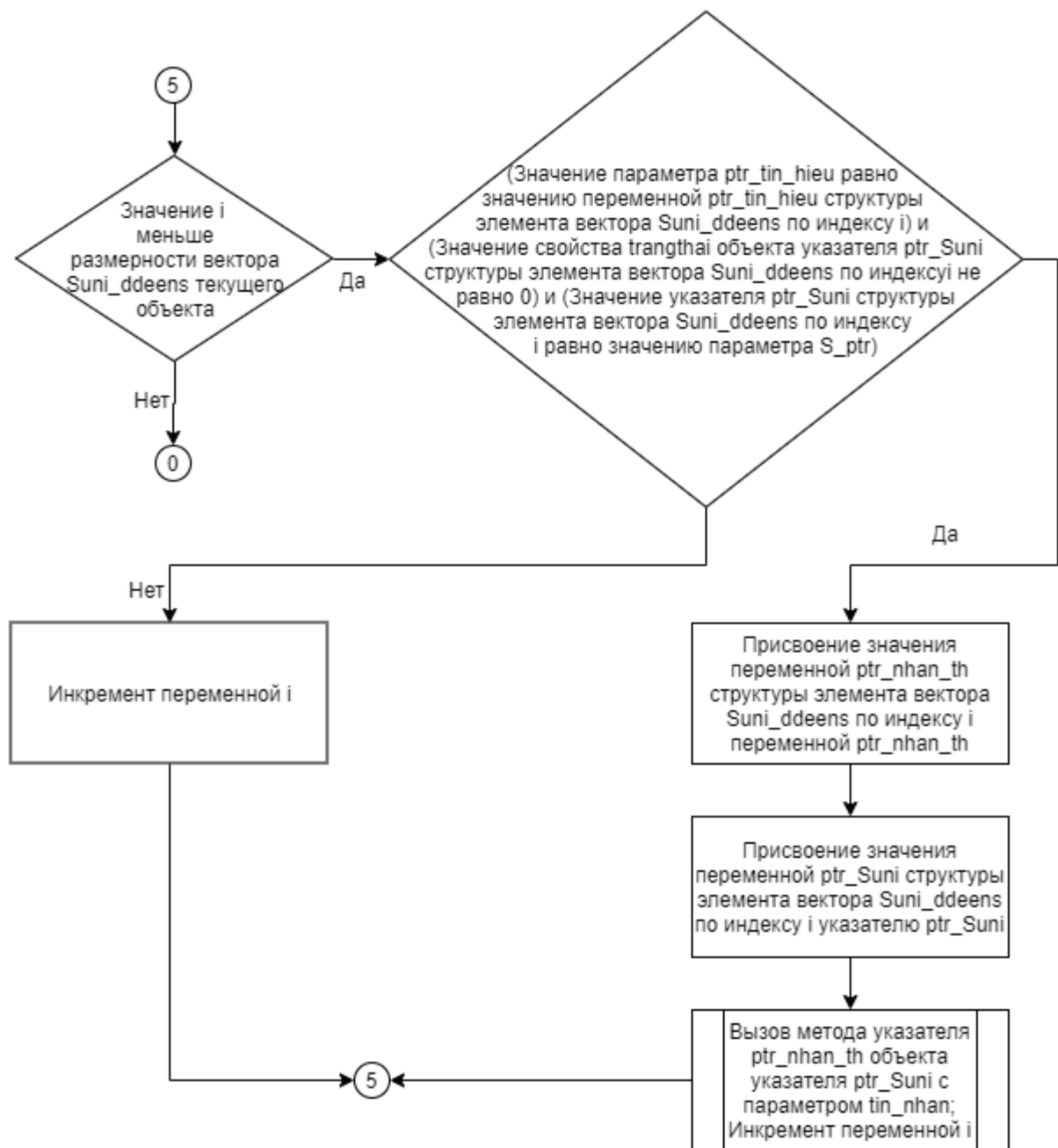


Рисунок 2 – Блок-схема алгоритма

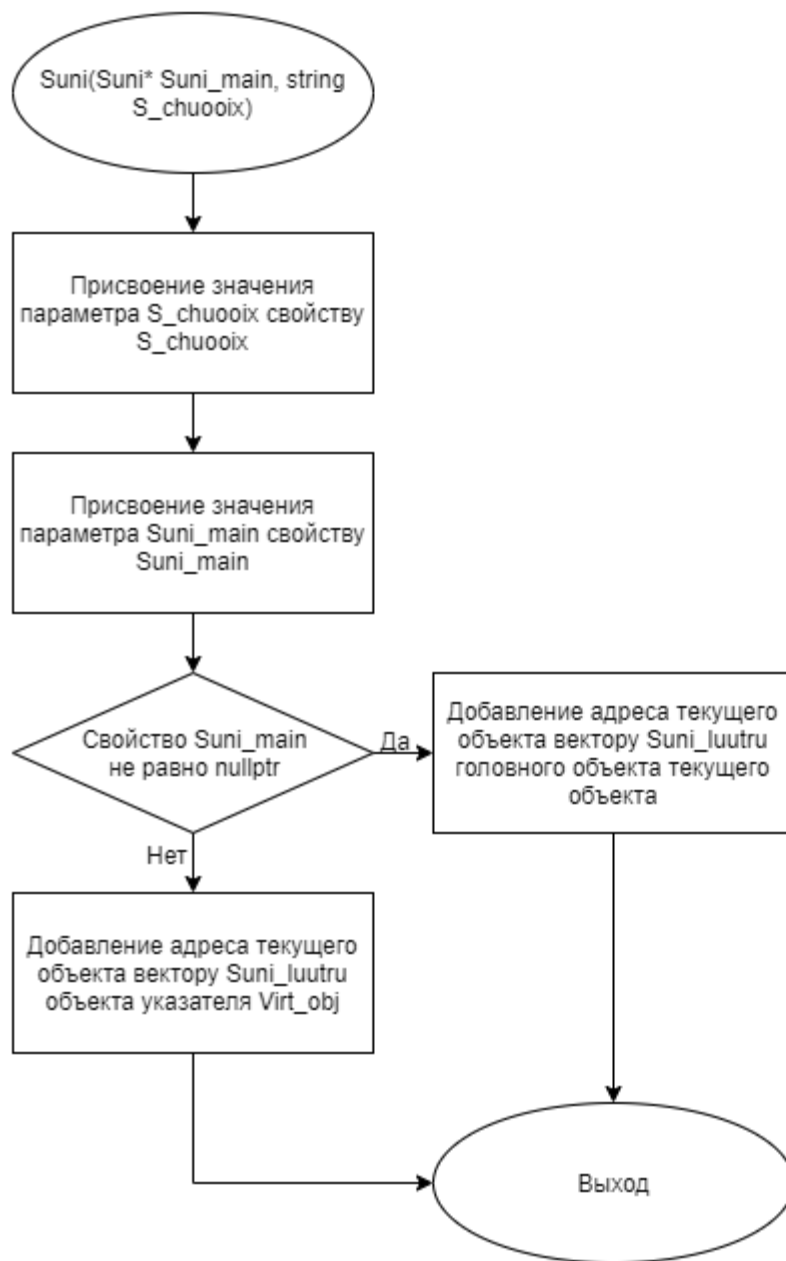


Рисунок 3 – Блок-схема алгоритма

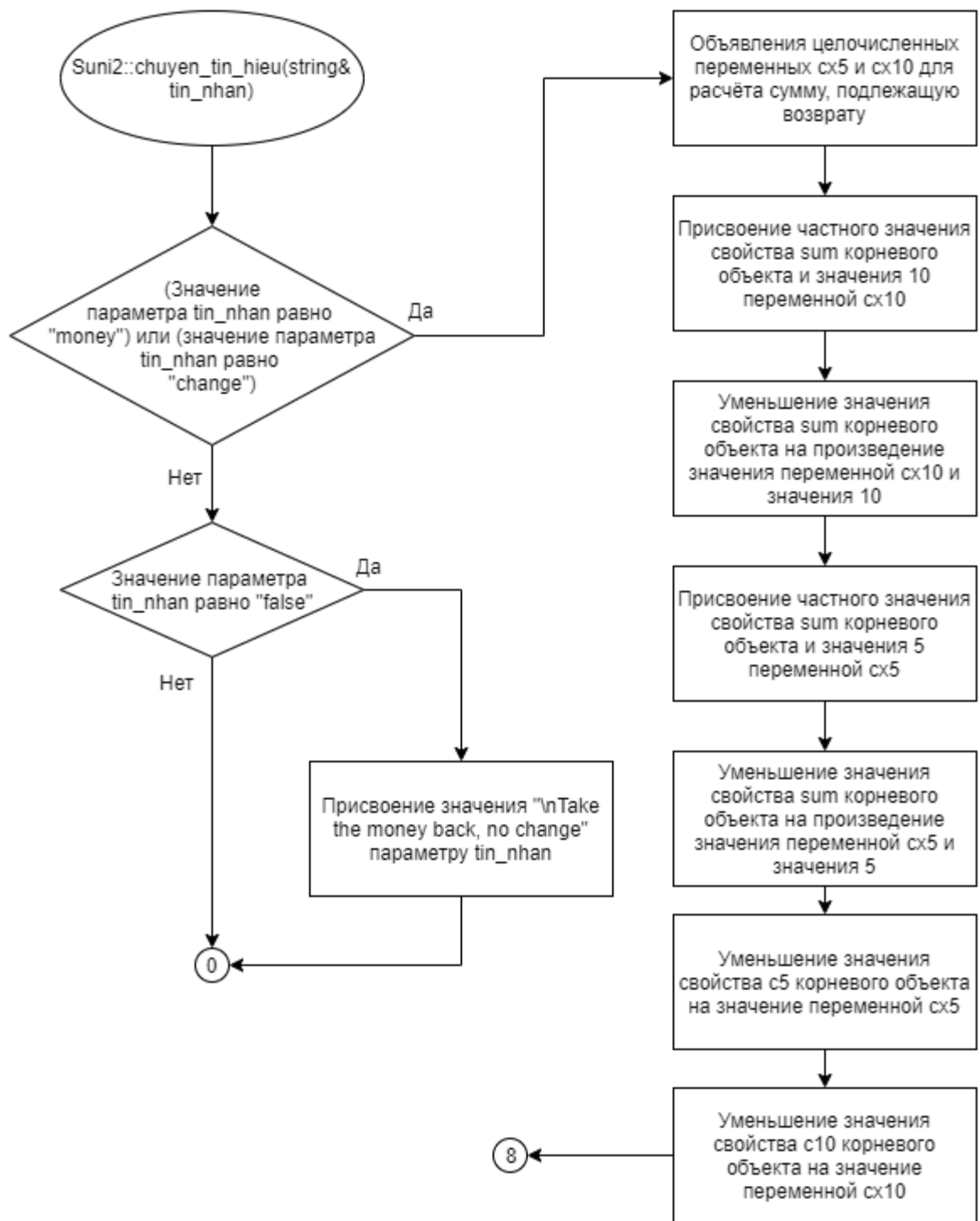


Рисунок 4 – Блок-схема алгоритма

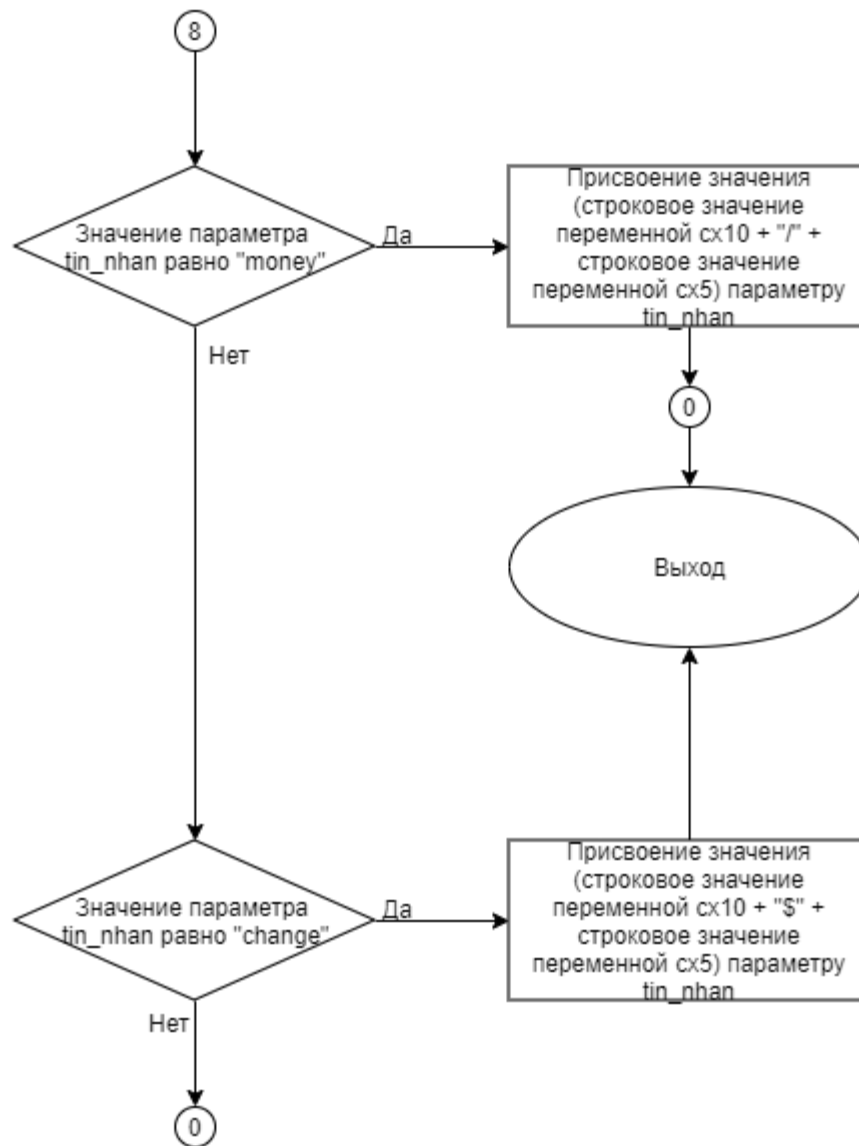


Рисунок 5 – Блок-схема алгоритма

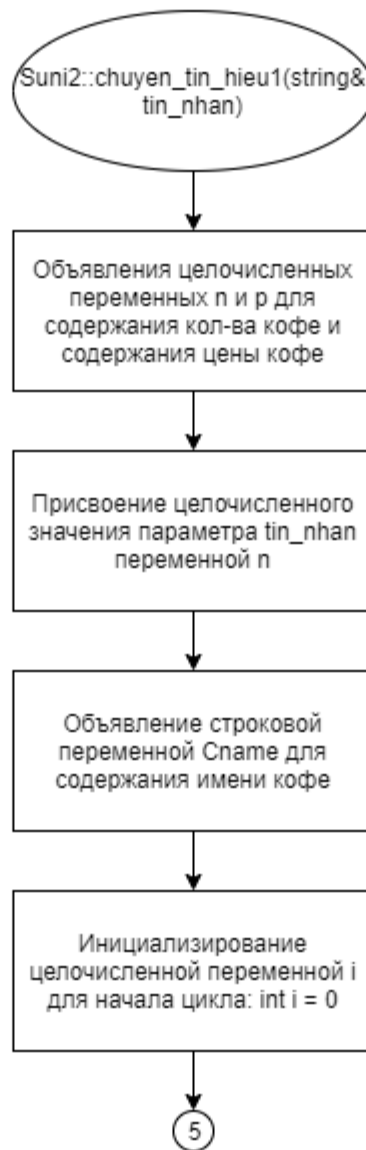


Рисунок 6 – Блок-схема алгоритма

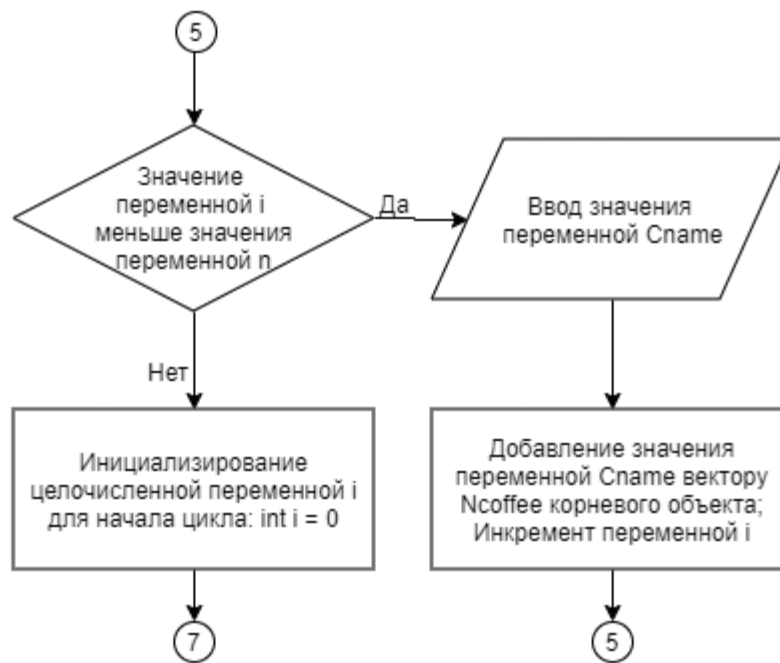


Рисунок 7 – Блок-схема алгоритма

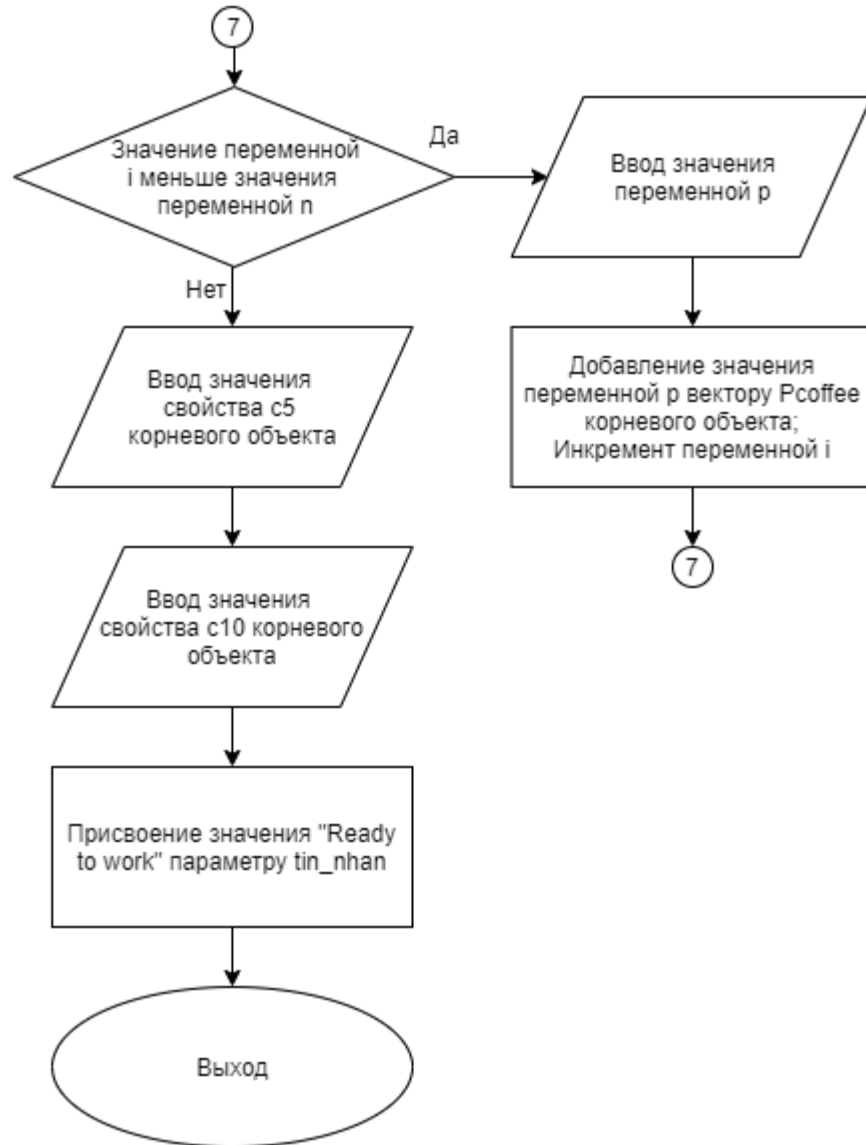


Рисунок 8 – Блок-схема алгоритма

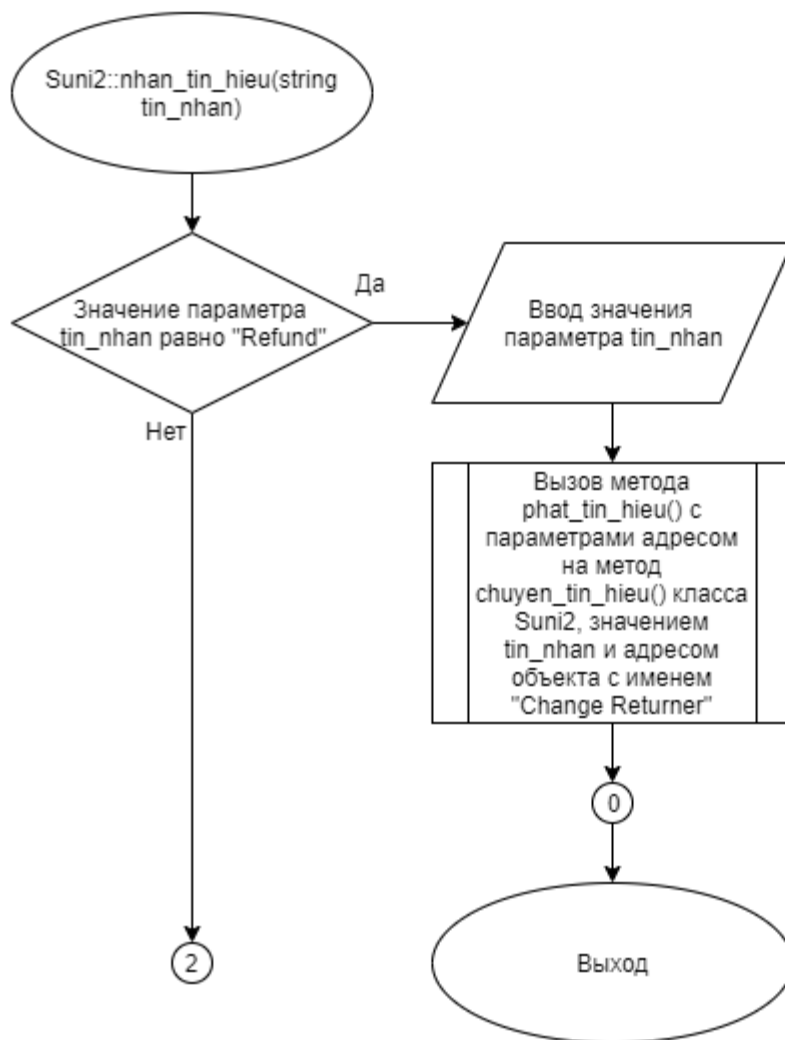


Рисунок 9 – Блок-схема алгоритма

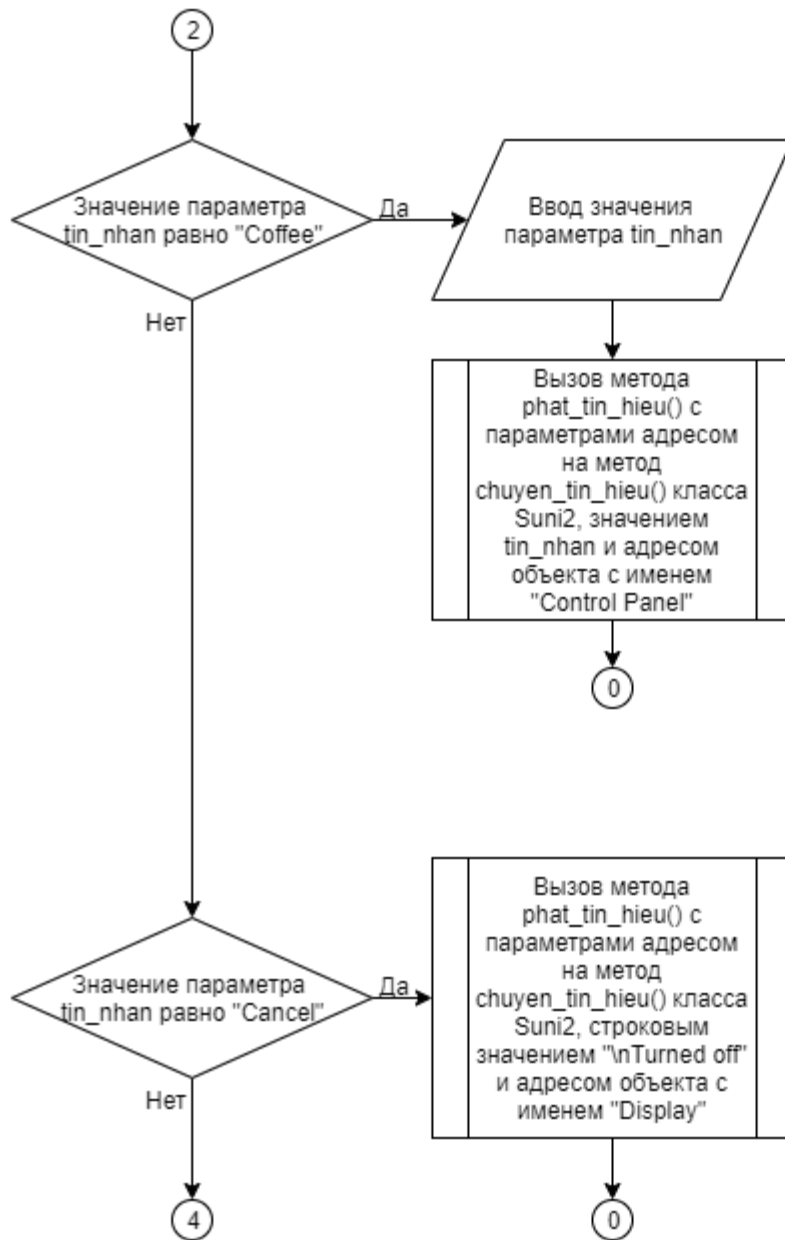


Рисунок 10 – Блок-схема алгоритма

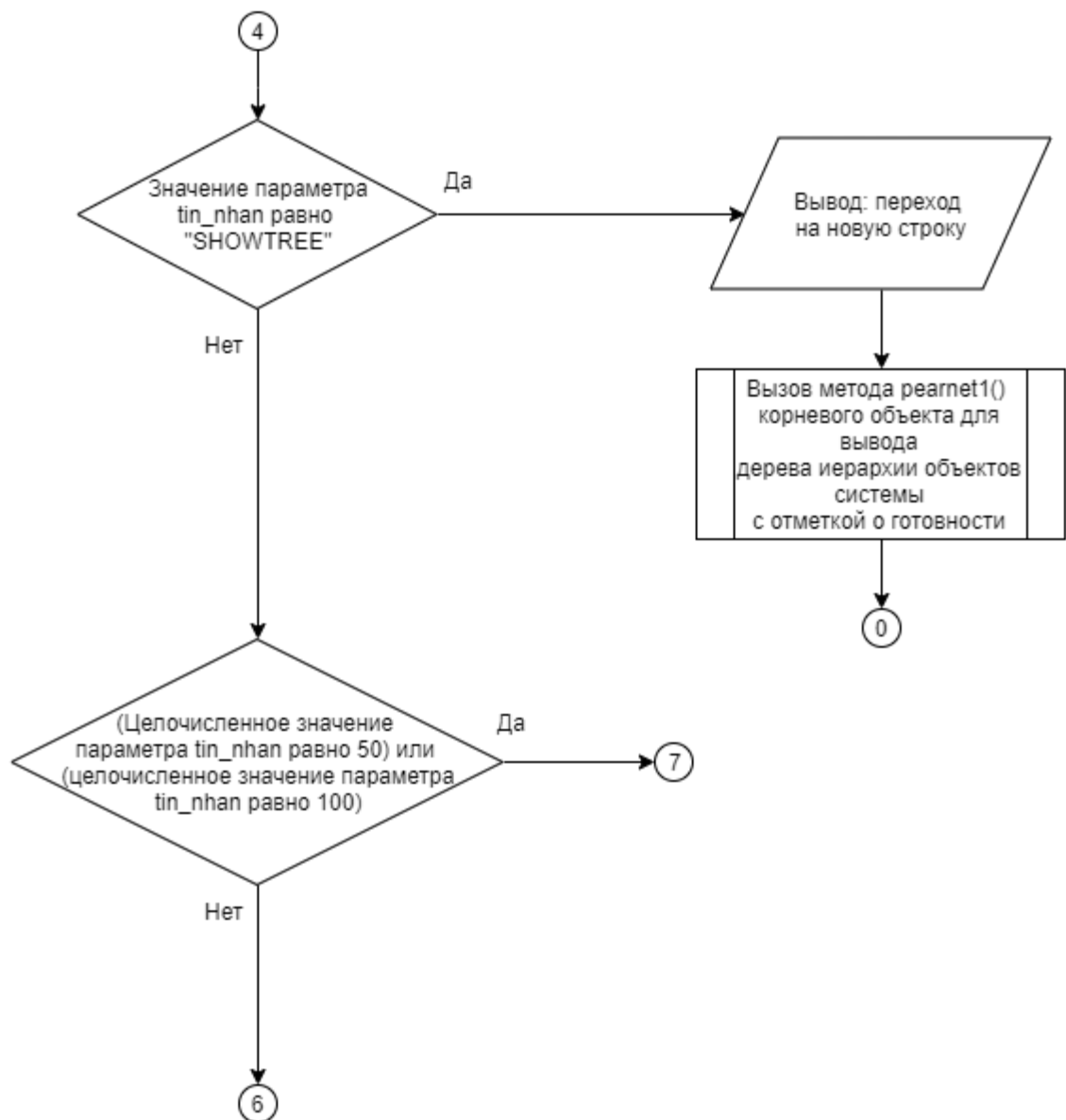


Рисунок 11 – Блок-схема алгоритма

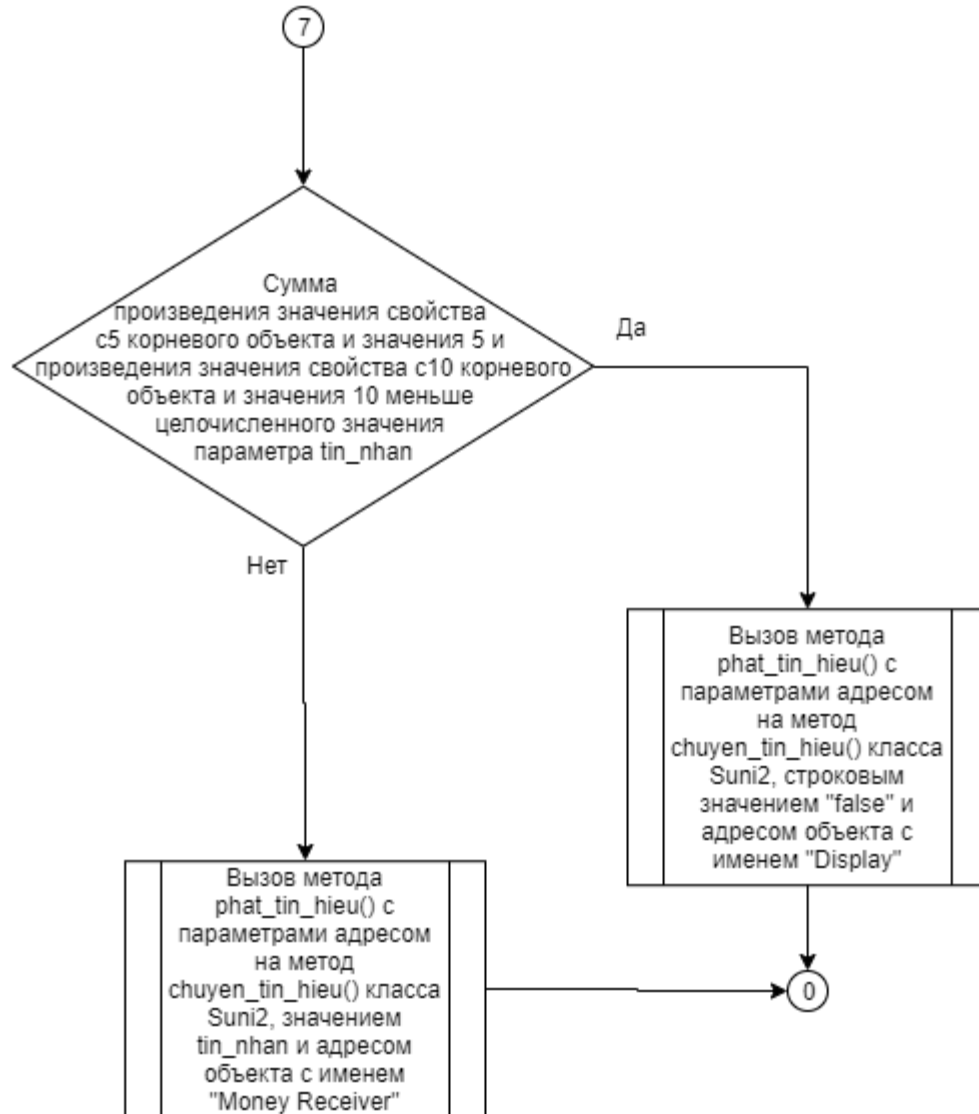


Рисунок 12 – Блок-схема алгоритма

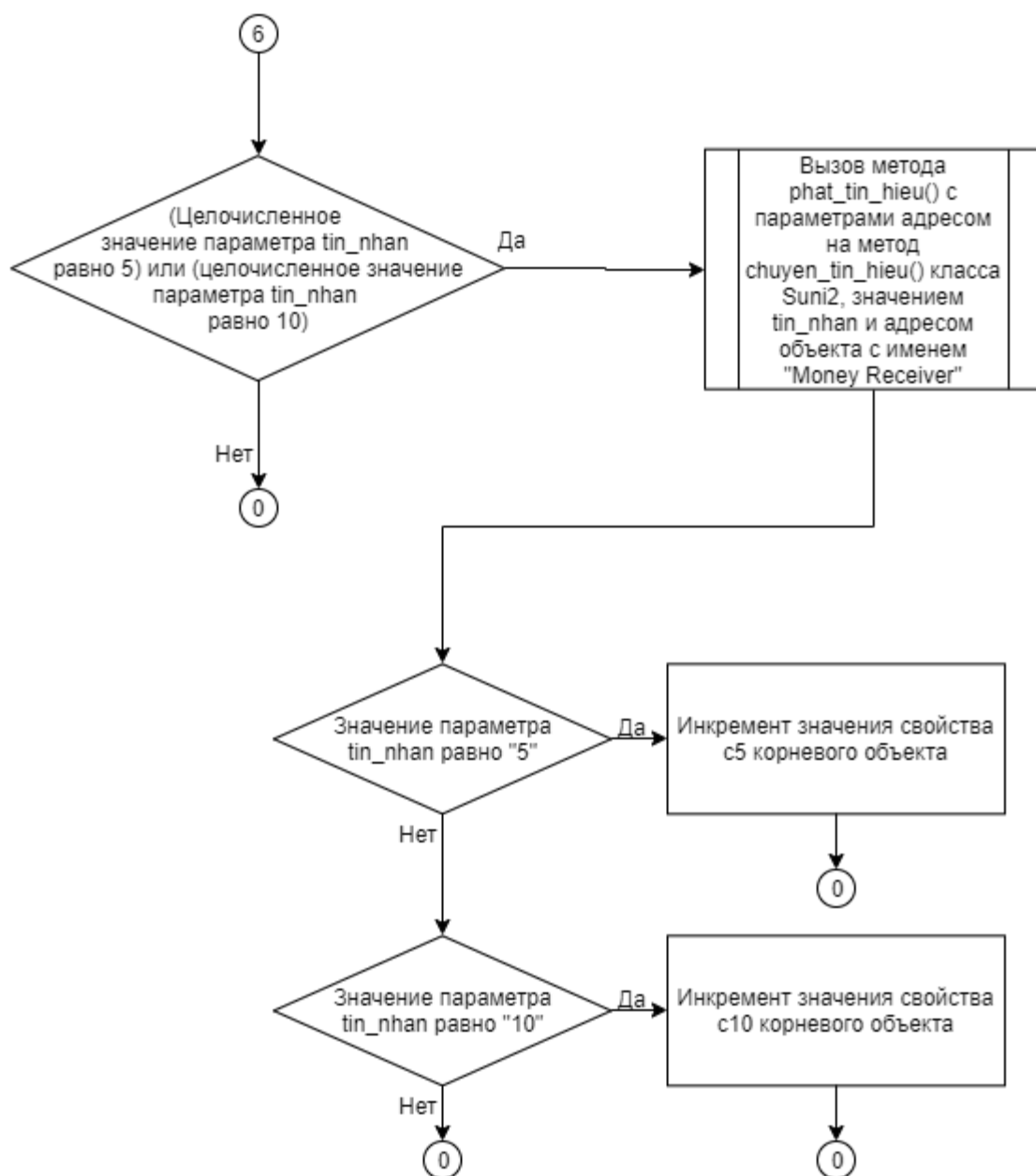


Рисунок 13 – Блок-схема алгоритма

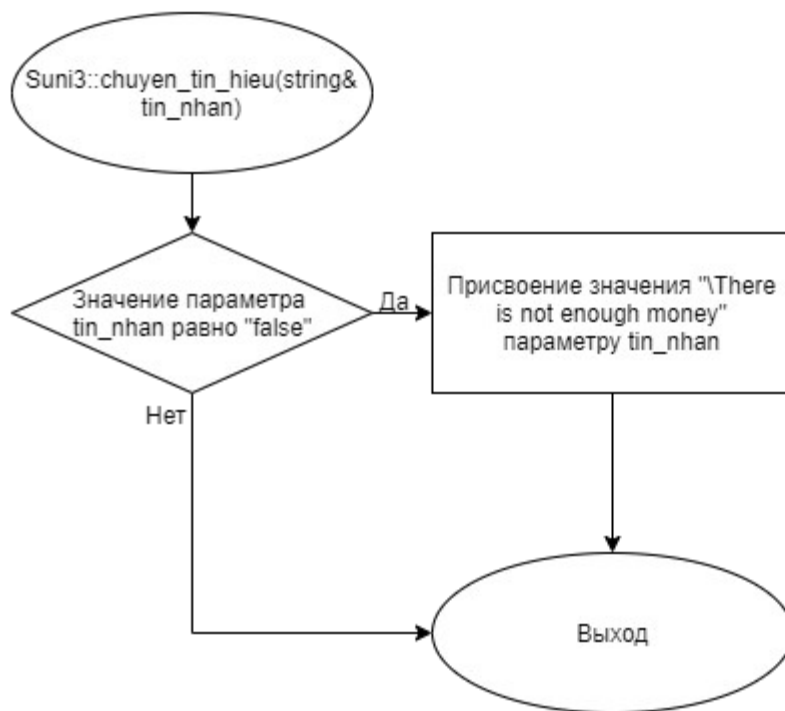


Рисунок 14 – Блок-схема алгоритма

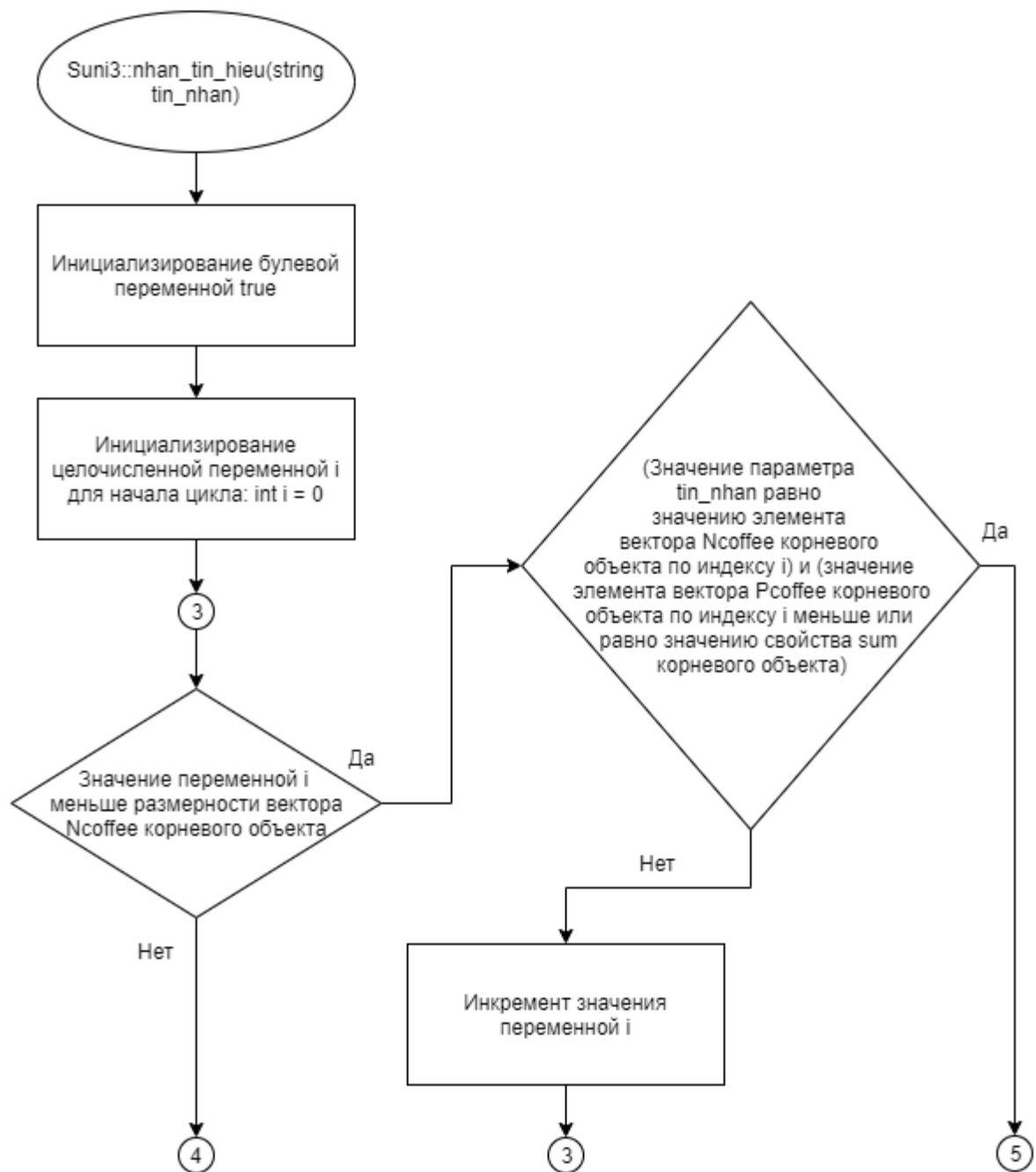


Рисунок 15 – Блок-схема алгоритма

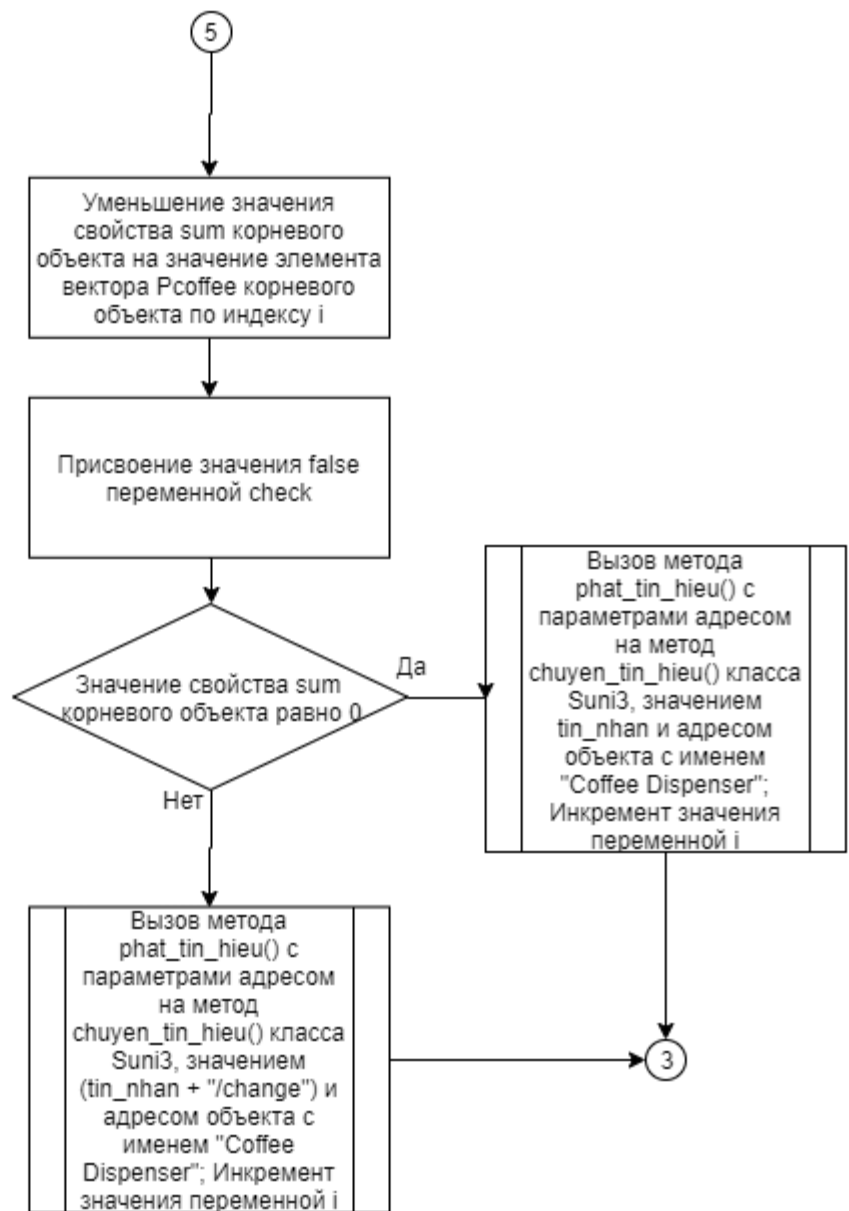


Рисунок 16 – Блок-схема алгоритма

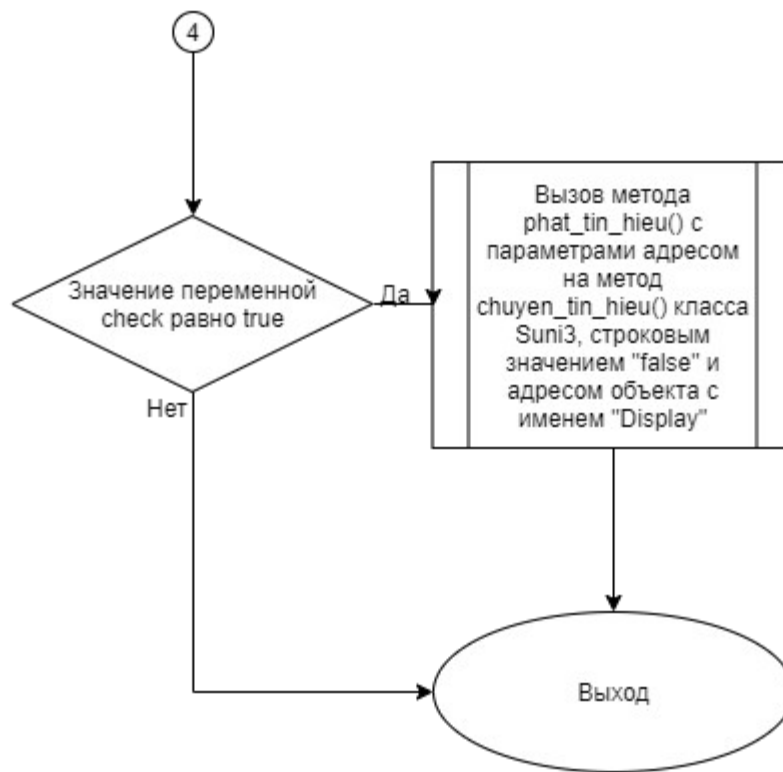


Рисунок 17 – Блок-схема алгоритма



Рисунок 18 – Блок-схема алгоритма

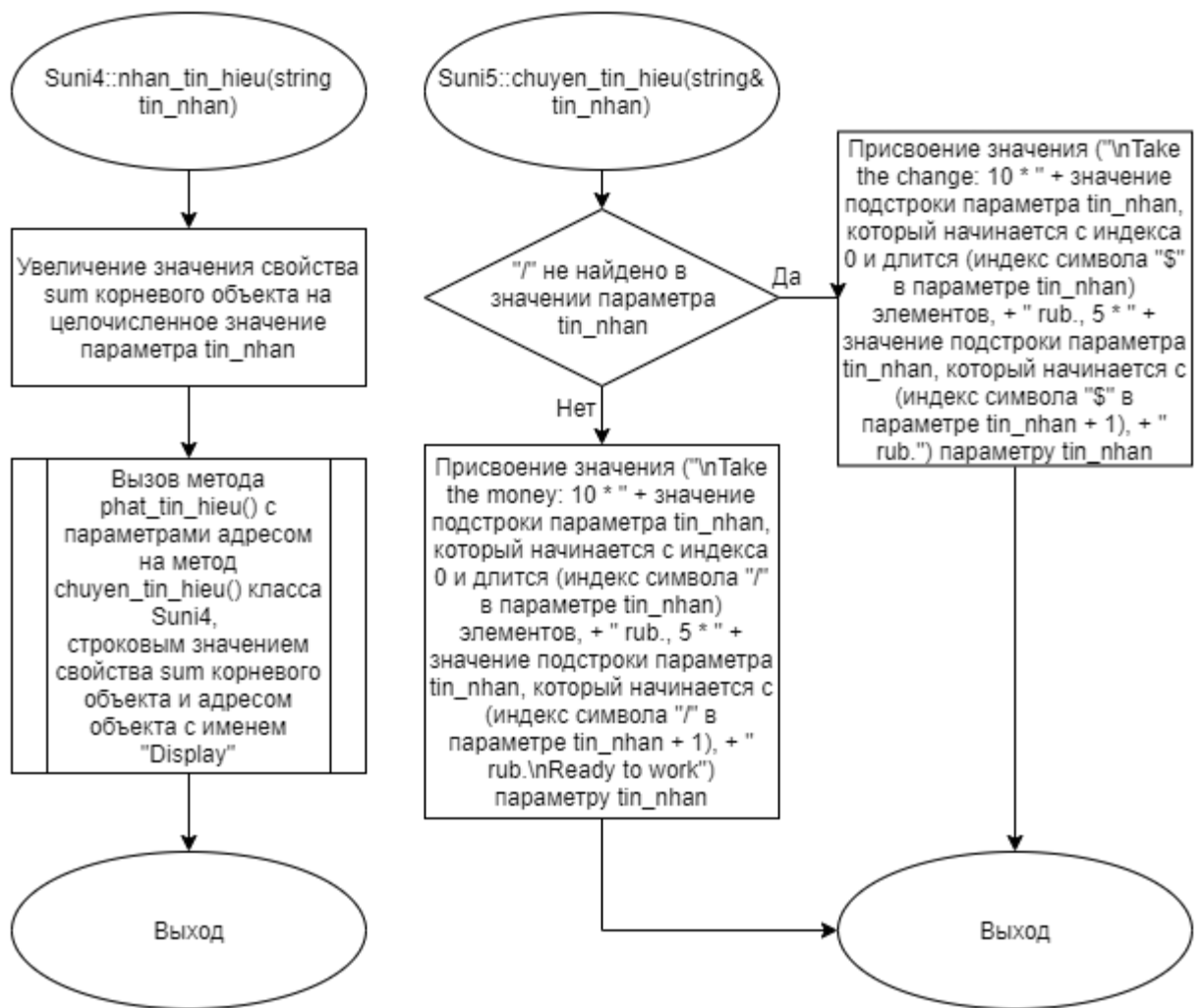


Рисунок 19 – Блок-схема алгоритма

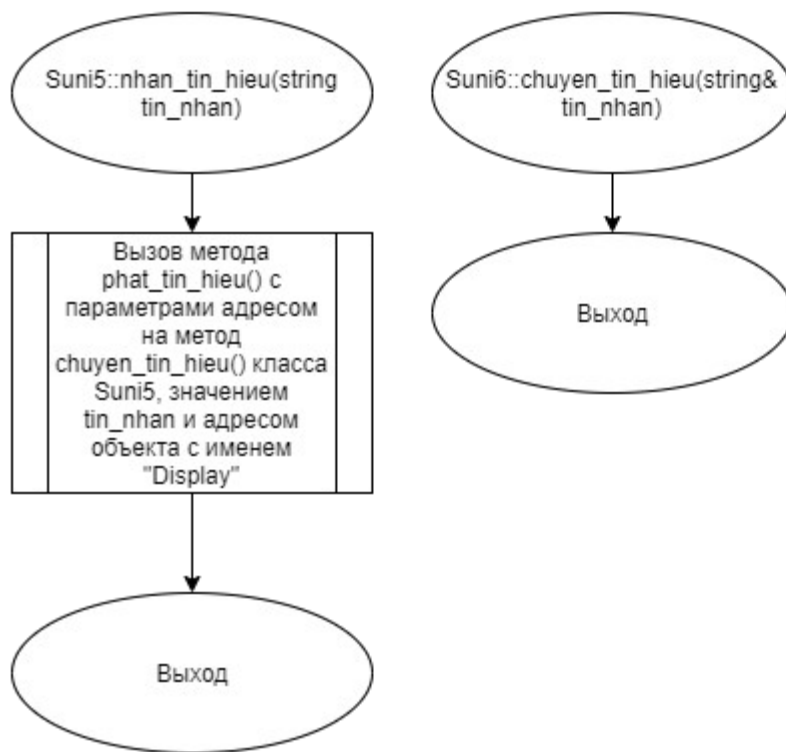


Рисунок 20 – Блок-схема алгоритма

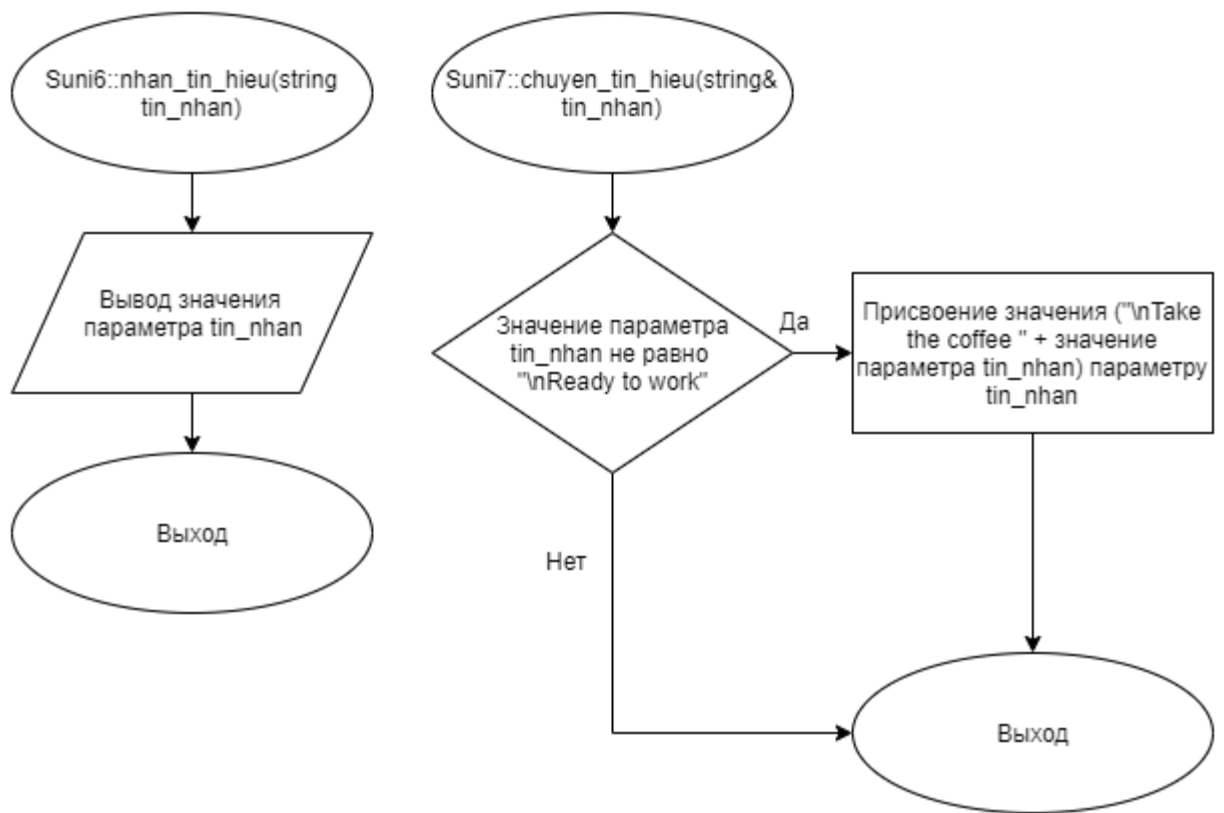


Рисунок 21 – Блок-схема алгоритма

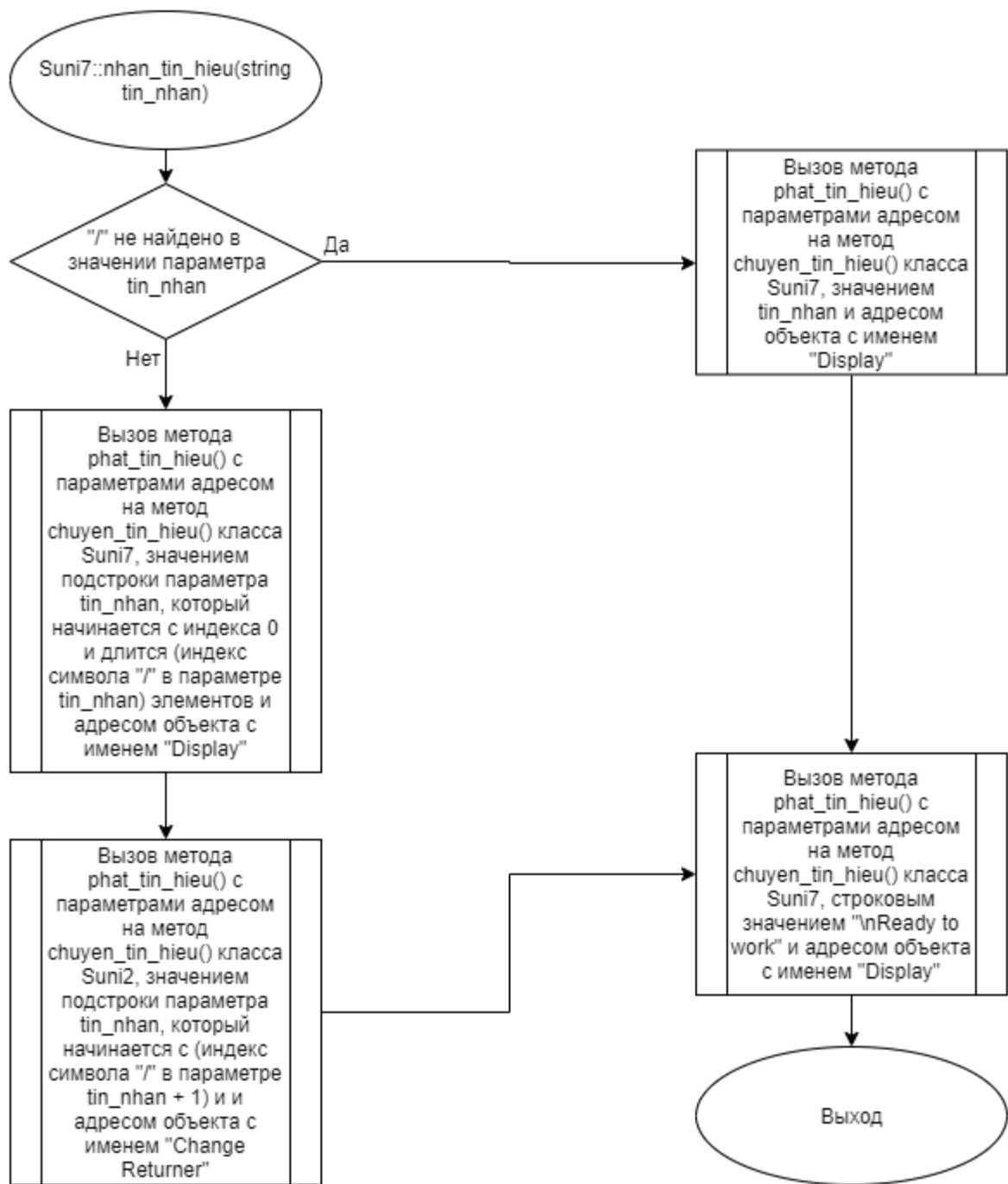


Рисунок 22 – Блок-схема алгоритма

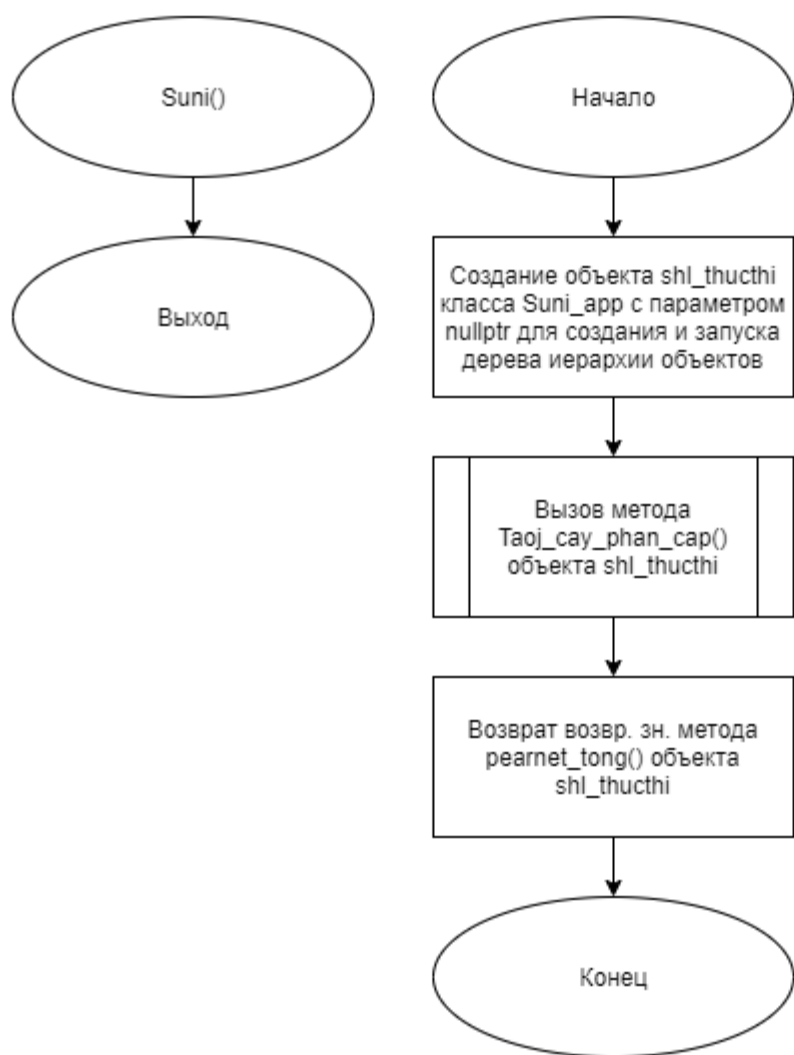


Рисунок 23 – Блок-схема алгоритма

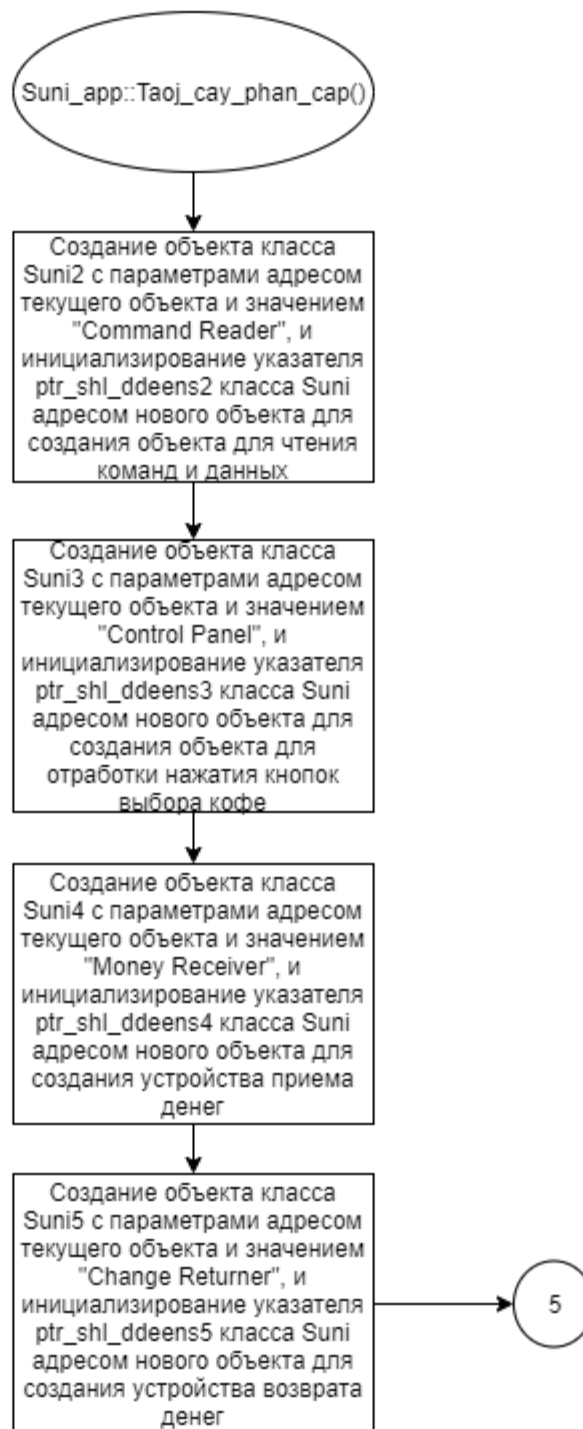


Рисунок 24 – Блок-схема алгоритма

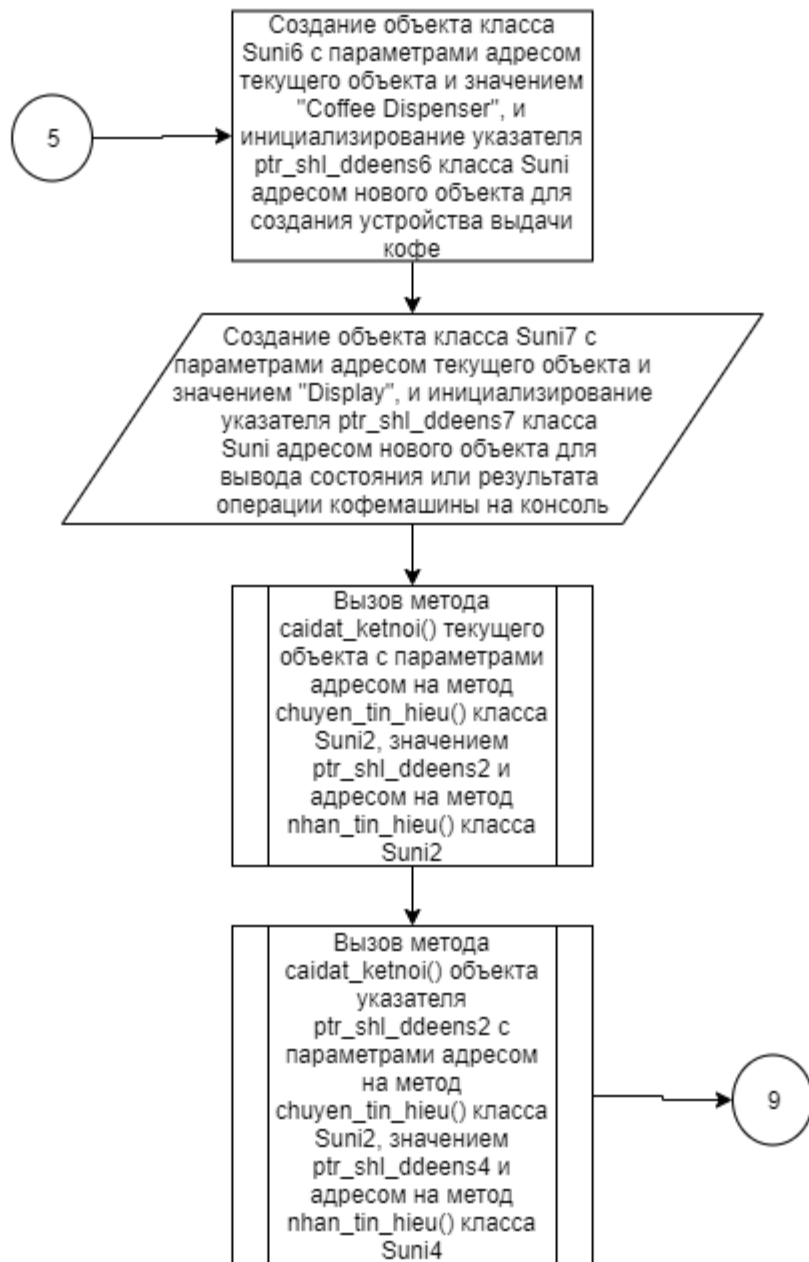


Рисунок 25 – Блок-схема алгоритма

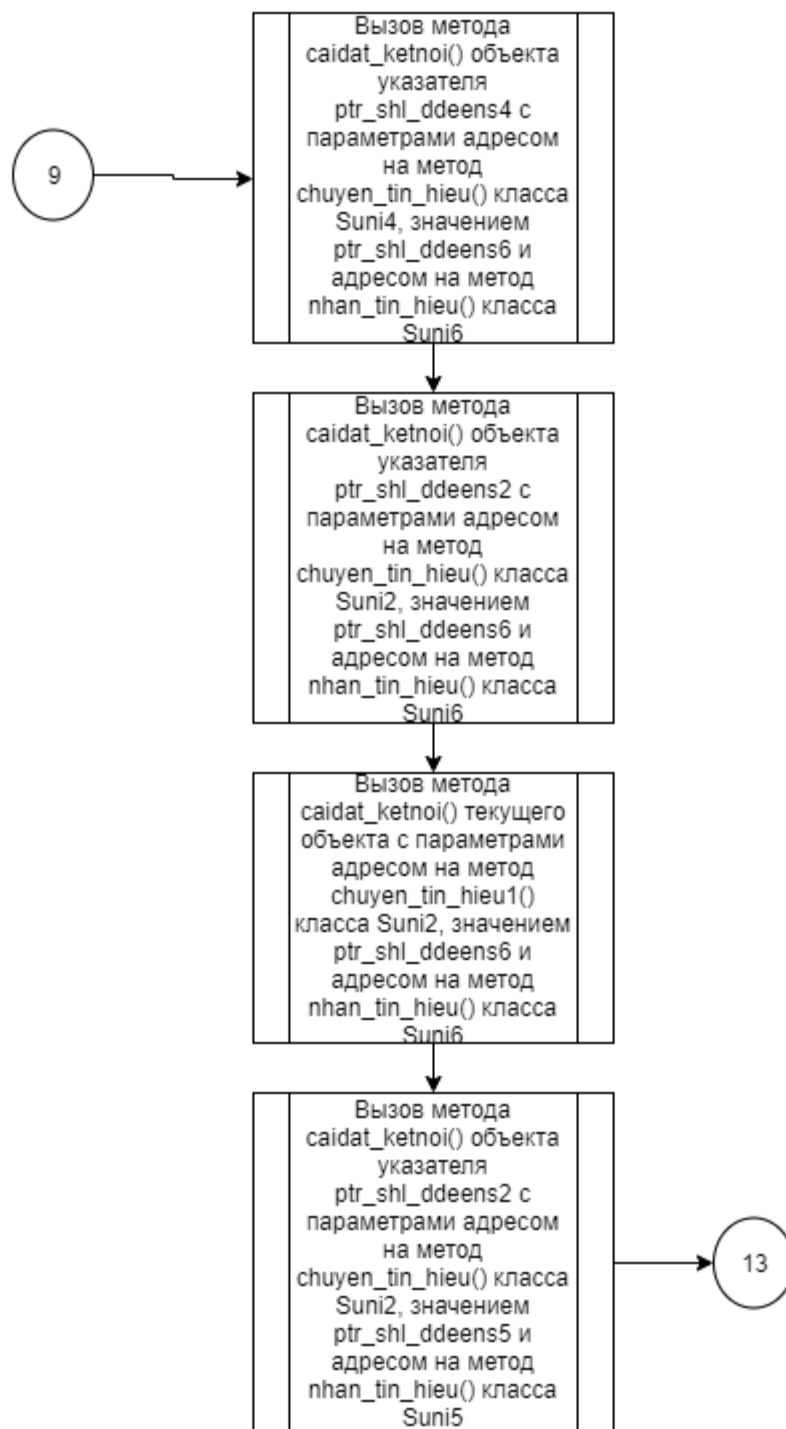


Рисунок 26 – Блок-схема алгоритма

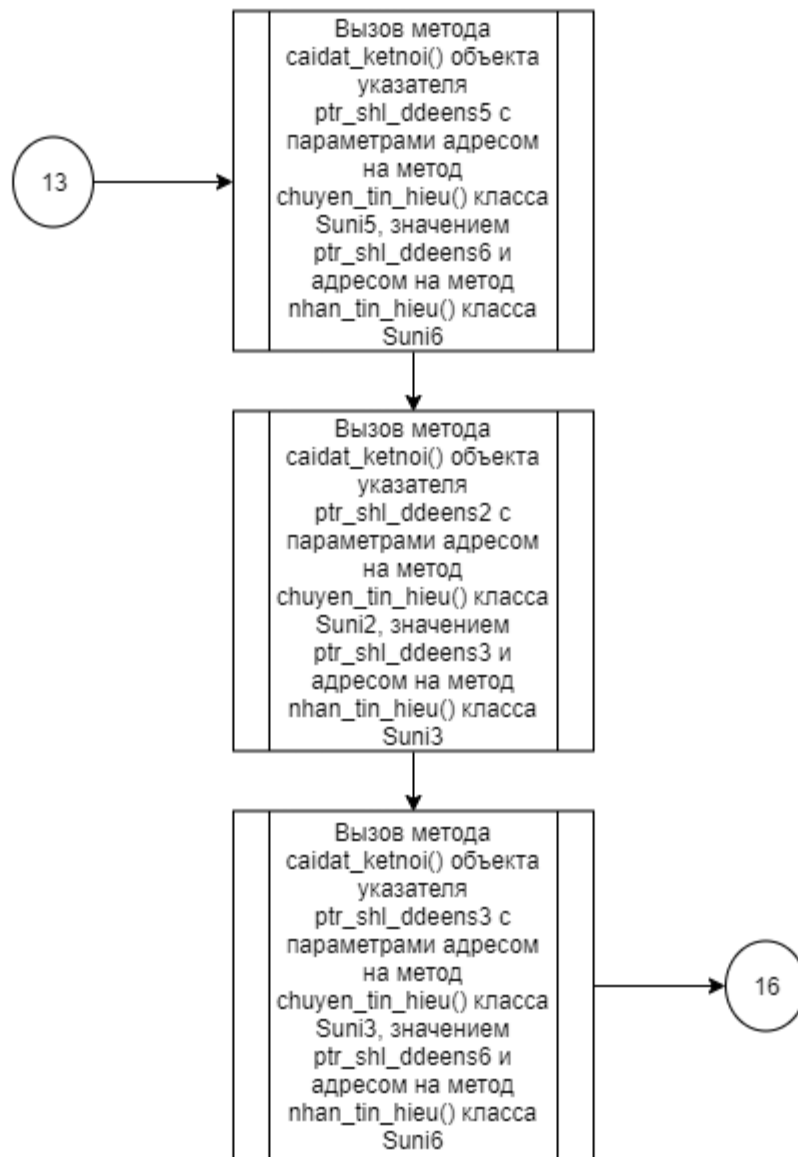


Рисунок 27 – Блок-схема алгоритма

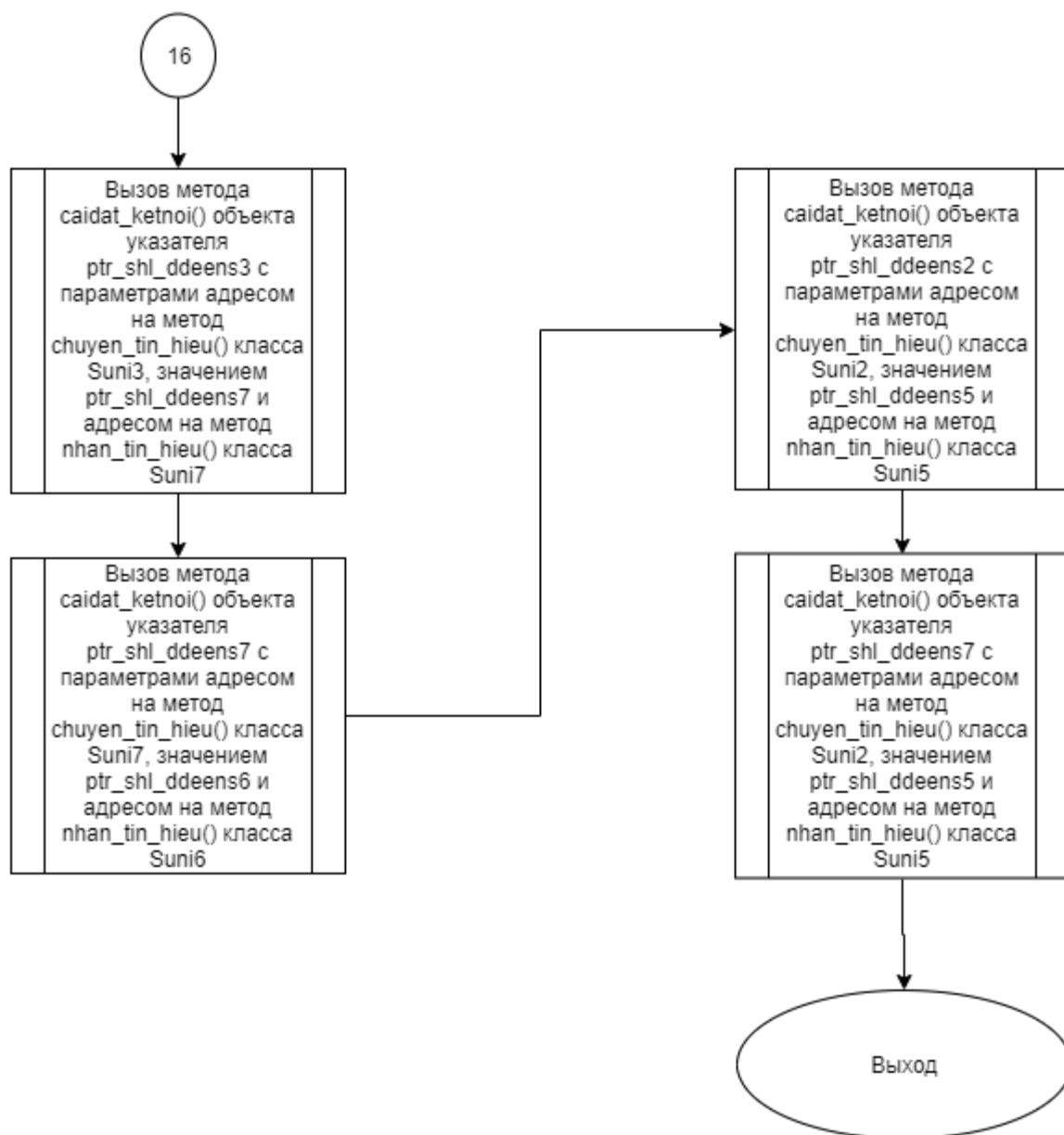


Рисунок 28 – Блок-схема алгоритма



Рисунок 29 – Блок-схема алгоритма



Рисунок 30 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "Suni_app.h"

int main()
{
    Suni_app shl_thucythi(nullptr);          // создание корневого объекта
    shl_thucythi.Taoj_cay_phan_cap();        // конструирование системы,
    построение дерева объектов
    return shl_thucythi.pearnet_tong();      // запуск системы
}
```

5.2 Файл Suni2.cpp

Листинг 2 – Suni2.cpp

```
#include "Suni2.h"

// Command Reader

Suni2::Suni2(Suni* Suni_main, string S_chuooix) : Suni(Suni_main, S_chuooix)
    // Вызов конструктора класса Suni
{} // Конструктор класса Suni2

// Метод возврата номера класса
int Suni2::Lay_solop() {
    return 2;
}

void Suni2::chuyen_tin_hieu(string& tin_nhan) {
    if (tin_nhan == "money" || tin_nhan == "change") {
        int cx5, cx10;
        cx10 = Virt_obj->Timf_object_from_current("System")->sum / 10;
        Virt_obj->Timf_object_from_current("System")->sum -= cx10 * 10;
        cx5 = Virt_obj->Timf_object_from_current("System")->sum / 5;
        Virt_obj->Timf_object_from_current("System")->sum -= cx5 * 5;
    }
}
```



```

        Virt_obj->Timf_object_from_current("System")->c5 -= cx5;
        Virt_obj->Timf_object_from_current("System")->c10 -= cx10;
        if (tin_nhan == "money")
            tin_nhan = to_string(cx10) + "/" + to_string(cx5);
        if (tin_nhan == "change")
            tin_nhan = to_string(cx10) + "$" + to_string(cx5);
    }

    if (tin_nhan == "false")
        tin_nhan = "\nTake the money back, no change";
}

void Suni2::nhan_tin_hieu(string tin_nhan) {
    if (tin_nhan == "Refund") {
        cin >> tin_nhan;
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),    tin_nhan,
Virt_obj->Timf_object_from_current("Change Returner"));
    }
    else if (tin_nhan == "Coffee") {
        cin >> tin_nhan;
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),    tin_nhan,
Virt_obj->Timf_object_from_current("Control Panel"));
    }
    else if (tin_nhan == "Cancel") {
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),    "\nTurned
off", Virt_obj->Timf_object_from_current("Display"));
    }
    else if (tin_nhan == "SHOWTREE") {
        cout << endl;
        Virt_obj->Timf_object_from_current("System")->pearnet1();    //    ...для
вывода дерева иерархии объектов системы с отметкой о готовности
    }
    else {
        if (stoi(tin_nhan) == 50 || stoi(tin_nhan) == 100) {
            if ((Virt_obj->Timf_object_from_current("System")->c5 * 5) +
(Virt_obj->Timf_object_from_current("System")->c10 * 10) < stoi(tin_nhan))
                this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),
"false", Virt_obj->Timf_object_from_current("Display"));
            else
                this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),
tin_nhan, Virt_obj->Timf_object_from_current("Money Receiver"));
        }
        else if (stoi(tin_nhan) == 5 || stoi(tin_nhan) == 10) {
            this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),
tin_nhan, Virt_obj->Timf_object_from_current("Money Receiver"));
            if (tin_nhan == "5")
                Virt_obj->Timf_object_from_current("System")->c5 += 1;
            if (tin_nhan == "10")
                Virt_obj->Timf_object_from_current("System")->c10 += 1;
        }
    }
}

void Suni2::chuyen_tin_hieu1(string& tin_nhan) {
    int n, p;

```

```

        n = stoi(tin_nhan);
        string Cname;
        for (int i = 0; i < n; i++) {
            cin >> Cname;
            Virt_obj->Timf_object_from_current("System")->Ncoffee.push_back(Cname);
        }
        for (int i = 0; i < n; i++) {
            cin >> p;
            Virt_obj->Timf_object_from_current("System")->Pcoffee.push_back(p);
        }

        cin >> Virt_obj->Timf_object_from_current("System")->c5;
        cin >> Virt_obj->Timf_object_from_current("System")->c10;
        tin_nhan = "Ready to work";
    }

    Suni2::~Suni2() {} // Деструктор класса Suni2

```

5.3 Файл Suni2.h

Листинг 3 – Suni2.h

```

#ifndef __SUNI2__H
#define __SUNI2__H

#include "Suni.h"

class Suni2: public Suni {
public:
    Suni2(Suni* Suni_main, string S_chuooix);
    int Lay_solop() override;
    void chuyen_tin_hieu(string& tin_nhan);
    void chuyen_tin_hieu1(string& tin_nhan);
    void nhan_tin_hieu(string tin_nhan);
    ~Suni2();
};

#endif

```

5.4 Файл Suni3.cpp

Листинг 4 – Suni3.cpp

```

#include "Suni3.h"

```

```

// Control Panel

Suni3::Suni3(Suni* Suni_main, string S_chuooix) : Suni(Suni_main, S_chuooix)
    // Вызов конструктора класса Suni
{} // Конструктор класса Suni3

// Метод возврата номера класса
int Suni3::Lay_solop() {
    return 3;
}

void Suni3::chuyen_tin_hieu(string& tin_nhan) {
    if (tin_nhan == "false")
        tin_nhan = "\nThere is not enough money";
}

void Suni3::nhan_tin_hieu(string tin_nhan) {
    bool check = true;
    for (int i = 0; i < Virt_obj->Timf_object_from_current("System")->Ncoffee.size(); i++) {
        if ((tin_nhan == Virt_obj->Timf_object_from_current("System")->Ncoffee[i])
            && (Virt_obj->Timf_object_from_current("System")->Pcoffee[i] <=
Virt_obj->Timf_object_from_current("System")->sum)) {
            Virt_obj->Timf_object_from_current("System")->sum -= Virt_obj->
Timf_object_from_current("System")->Pcoffee[i];
            check = false;
            if (Virt_obj->Timf_object_from_current("System")->sum == 0)
                this->phat_tin_hieu((TYPE_SIGNAL)(&Suni3::chuyen_tin_hieu),
tin_nhan, Virt_obj->Timf_object_from_current("Coffee Dispenser"));
            else
                this->phat_tin_hieu((TYPE_SIGNAL)(&Suni3::chuyen_tin_hieu),
tin_nhan + "/change", Virt_obj->Timf_object_from_current("Coffee
Dispenser"));
        }
    }
    if (check)
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni3::chuyen_tin_hieu), "false",
Virt_obj->Timf_object_from_current("Display"));
}

Suni3::~~Suni3() {} // Деструктор класса Suni3

```

5.5 Файл Suni3.h

Листинг 5 – Suni3.h

```

#ifndef __SUNI3__H
#define __SUNI3__H

```

```

#include "Suni.h"

class Suni3: public Suni {
public:
    Suni3(Suni* Suni_main, string S_chuooix);
    int Lay_solop() override;
    void chuyen_tin_hieu(string& tin_nhan);
    void nhan_tin_hieu(string tin_nhan);
    ~Suni3();
};

#endif

```

5.6 Файл Suni4.cpp

Листинг 6 – Suni4.cpp

```

#include "Suni4.h"

// Money Receiver

Suni4::Suni4(Suni* Suni_main, string S_chuooix) : Suni(Suni_main, S_chuooix)
    // Вызов конструктора класса Suni
{} // Конструктор класса Suni4

// Метод возврата номера класса
int Suni4::Lay_solop() {
    return 4;
}

void Suni4::chuyen_tin_hieu(string& tin_nhan) {
    tin_nhan = "\nThe amount: " + tin_nhan;
}

void Suni4::nhan_tin_hieu(string tin_nhan) {
    Virt_obj->Timf_object_from_current("System") -> sum += stoi(tin_nhan);
    phat_tin_hieu((TYPE_SIGNAL)(&Suni4::chuyen_tin_hieu), to_string(Virt_obj -
> Timf_object_from_current("System") -> sum),
    Virt_obj->Timf_object_from_current("Display"));
}

Suni4::~~Suni4() {} // Деструктор класса Suni4

```

5.7 Файл Suni4.h

Листинг 7 – Suni4.h

```
#ifndef __SUNI4__H
#define __SUNI4__H

#include "Suni.h"

class Suni4: public Suni {
public:
    Suni4(Suni* Suni_main, string S_chuooix);
    int Lay_solop() override;
    void chuyen_tin_hieu(string& tin_nhan);
    void nhan_tin_hieu(string tin_nhan);
    ~Suni4();
};

#endif
```

5.8 Файл Suni5.cpp

Листинг 8 – Suni5.cpp

```
#include "Suni5.h"

// Change Returner

Suni5::Suni5(Suni* Suni_main, string S_chuooix) : Suni(Suni_main, S_chuooix)
    // Вызов конструктора класса Suni
{} // Конструктор класса Suni5

// Метод возврата номера класса
int Suni5::Lay_solop() {
    return 5;
}

void Suni5::chuyen_tin_hieu(string& tin_nhan) {
    if (tin_nhan.find("/") == -1)
        tin_nhan = "\nTake the change: 10 * " + tin_nhan.substr(0,
tin_nhan.find("$")) + " rub., 5 * " + tin_nhan.substr(tin_nhan.find("$") +
1) + " rub.";
    else
        tin_nhan = "\nTake the money: 10 * " + tin_nhan.substr(0,
tin_nhan.find("/")) + " rub., 5 * " + tin_nhan.substr(tin_nhan.find("/") +
1) + " rub.\nReady to work";
}
```

```

void Suni5::nhan_tin_hieu(string tin_nhan) {
    this->phat_tin_hieu((TYPE_SIGNAL)&Suni5::chuyen_tin_hieu,      tin_nhan,
Virt_obj->Timf_object_from_current("Display"));
}

Suni5::~~Suni5() {} // Деструктор класса Suni5

```

5.9 Файл Suni5.h

Листинг 9 – Suni5.h

```

#ifndef __SUNI5__H
#define __SUNI5__H

#include "Suni.h"

class Suni5: public Suni {
public:
    Suni5(Suni* Suni_main, string S_chuooix);
    int Lay_solop() override;
    void chuyen_tin_hieu(string& tin_nhan);
    void nhan_tin_hieu(string tin_nhan);
    ~Suni5();
};

#endif

```

5.10 Файл Suni6.cpp

Листинг 10 – Suni6.cpp

```

#include "Suni6.h"

// Display

Suni6::Suni6(Suni* Suni_main, string S_chuooix) : Suni(Suni_main, S_chuooix)
    // Вызов конструктора класса Suni
{} // Конструктор класса Suni6

// Метод возврата номера класса
int Suni6::Lay_solop() {
    return 6;
}

void Suni6::chuyen_tin_hieu(string&) {}

```

```

void Suni6::nhan_tin_hieu(string tin_nhan) {
    cout << tin_nhan;
}

Suni6::~~Suni6() {} // Деструктор класса Suni6

```

5.11 Файл Suni6.h

Листинг 11 – Suni6.h

```

#ifndef __SUNI6__H
#define __SUNI6__H

#include "Suni.h"

class Suni6: public Suni {
public:
    Suni6(Suni* Suni_main, string S_chuooix);
    int Lay_solop() override;
    void chuyen_tin_hieu(string& tin_nhan);
    void nhan_tin_hieu(string tin_nhan);
    ~Suni6();
};

#endif

```

5.12 Файл Suni7.cpp

Листинг 12 – Suni7.cpp

```

#include "Suni2.h"
#include "Suni7.h"
#include "Suni_app.h"

// Coffee Dispenser

Suni7::Suni7(Suni* Suni_main, string S_chuooix) : Suni(Suni_main, S_chuooix)
    // Вызов конструктора класса Suni
{} // Конструктор класса Suni7

// Метод возврата номера класса
int Suni7::Lay_solop() {
    return 7;
}

```

```

void Suni7::chuyen_tin_hieu(string& tin_nhan) {
    if (tin_nhan != "\nReady to work")
        tin_nhan = "\nTake the coffee " + tin_nhan;
}

void Suni7::nhan_tin_hieu(string tin_nhan) {
    if (tin_nhan.find("/") == -1)
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni7::chuyen_tin_hieu),    tin_nhan,
Virt_obj->Timf_object_from_current("Display"));
    else {
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni7::chuyen_tin_hieu),
tin_nhan.substr(0,          tin_nhan.find("/")),          Virt_obj-
>Timf_object_from_current("Display"));
        this->phat_tin_hieu((TYPE_SIGNAL)(&Suni2::chuyen_tin_hieu),
tin_nhan.substr(tin_nhan.find("/")          +          1),          Virt_obj-
>Timf_object_from_current("Change Returner"));
    }
    this->phat_tin_hieu((TYPE_SIGNAL)(&Suni7::chuyen_tin_hieu),    "\nReady to
work", Virt_obj->Timf_object_from_current("Display"));
}

Suni7::~~Suni7() {} // Деструктор класса Suni7

```

5.13 Файл Suni7.h

Листинг 13 – Suni7.h

```

#ifndef __SUNI7__H
#define __SUNI7__H

#include "Suni.h"

class Suni7 : public Suni {
public:
    Suni7(Suni* Suni_main, string S_chuooix);
    int Lay_solop() override;
    void chuyen_tin_hieu(string&);
    void nhan_tin_hieu(string);
    ~Suni7();
};

#endif

```


5.14 Файл Suni_app.cpp

Листинг 14 – Suni_app.cpp

```
#include "Suni_app.h"
#include "Suni2.h"
#include "Suni3.h"
#include "Suni4.h"
#include "Suni5.h"
#include "Suni6.h"
#include "Suni7.h"

// Конструктор класса Suni_app
Suni_app::Suni_app(Suni* Suni_main) : Suni(Suni_main) {}

// Метод создания дерева иерархии
void Suni_app::Taoj_cay_phan_cap() {
    // Построение дерева иерархии объектов
    Suni* ptr_shl_ddeens2 = new Suni2(this, "Command Reader");
    Suni* ptr_shl_ddeens3 = new Suni3(this, "Control Panel");
    Suni* ptr_shl_ddeens4 = new Suni4(this, "Money Receiver");
    Suni* ptr_shl_ddeens5 = new Suni5(this, "Change Returner");
    Suni* ptr_shl_ddeens6 = new Suni6(this, "Display");
    Suni* ptr_shl_ddeens7 = new Suni7(this, "Coffee Dispenser");

    // Установка связей сигналов и обработчиков между объектами
    caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu, ptr_shl_ddeens2,
        (TYPE_HANDLER)&Suni2::nhan_tin_hieu));
    ptr_shl_ddeens2->caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu,
        ptr_shl_ddeens4, (TYPE_HANDLER)&Suni4::nhan_tin_hieu));
    ptr_shl_ddeens4->caidat_ketnoi((TYPE_SIGNAL)&Suni4::chuyen_tin_hieu,
        ptr_shl_ddeens6, (TYPE_HANDLER)&Suni6::nhan_tin_hieu));
    ptr_shl_ddeens2->caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu,
        ptr_shl_ddeens6, (TYPE_HANDLER)&Suni6::nhan_tin_hieu));
    caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu1, ptr_shl_ddeens6,
        (TYPE_HANDLER)&Suni6::nhan_tin_hieu));
    ptr_shl_ddeens2->caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu,
        ptr_shl_ddeens5, (TYPE_HANDLER)&Suni5::nhan_tin_hieu));
    ptr_shl_ddeens5->caidat_ketnoi((TYPE_SIGNAL)&Suni5::chuyen_tin_hieu,
        ptr_shl_ddeens6, (TYPE_HANDLER)&Suni6::nhan_tin_hieu));
    ptr_shl_ddeens2->caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu,
        ptr_shl_ddeens3, (TYPE_HANDLER)&Suni3::nhan_tin_hieu));
    ptr_shl_ddeens3->caidat_ketnoi((TYPE_SIGNAL)&Suni3::chuyen_tin_hieu,
        ptr_shl_ddeens6, (TYPE_HANDLER)&Suni6::nhan_tin_hieu));
    ptr_shl_ddeens3->caidat_ketnoi((TYPE_SIGNAL)&Suni3::chuyen_tin_hieu,
        ptr_shl_ddeens7, (TYPE_HANDLER)&Suni7::nhan_tin_hieu));
    ptr_shl_ddeens7->caidat_ketnoi((TYPE_SIGNAL)&Suni7::chuyen_tin_hieu,
        ptr_shl_ddeens6, (TYPE_HANDLER)&Suni6::nhan_tin_hieu));
    ptr_shl_ddeens2->caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu,
        ptr_shl_ddeens5, (TYPE_HANDLER)&Suni5::nhan_tin_hieu));
    ptr_shl_ddeens7->caidat_ketnoi((TYPE_SIGNAL)&Suni2::chuyen_tin_hieu,
        ptr_shl_ddeens5, (TYPE_HANDLER)&Suni5::nhan_tin_hieu));
}
```

```

// Метод запуска системы
int Suni_app::pearnet_tong() {
    // Приведение всех объектов в состояние готовности
    Virt_obj -> Active();

    string command;
    getline(cin, command, ' ');

    Virt_obj -> Timf_object_from_current("System") ->
    phat_tin_hieu((TYPE_SIGNAL>(&Suni2::chuyen_tin_hieu1), command, Virt_obj-
    >Timf_object_from_current("Display"));

    do {
        cin >> command;
        Virt_obj->Timf_object_from_current("System")-
    >phat_tin_hieu((TYPE_SIGNAL>(&Suni2::chuyen_tin_hieu), command, Virt_obj-
    >Timf_object_from_current("Command Reader"));
        } while (command != "Cancel" && command != "SHOWTREE");

    return(0);
}

Suni_app::~Suni_app() {}

```

5.15 Файл Suni_app.h

Листинг 15 – Suni_app.h

```

#ifndef __SUNI_APP__H
#define __SUNI_APP__H

#include "Suni.h"
#include "Suni2.h"
#include "Suni3.h"
#include "Suni4.h"
#include "Suni5.h"
#include "Suni6.h"
#include "Suni7.h"

class Suni_app: public Suni {
public:
    Suni_app(Suni* Suni_main);
    void Taoj_cay_phan_cap();
    int pearnet_tong();
    ~Suni_app();
};

#endif

```

5.16 Файл Suni.cpp

Листинг 16 – Suni.cpp

```
#include "Suni.h"

// Конструктор класса Suni по умолчанию
Suni::Suni() {}

// Конструктор класса Suni с параметрами
Suni::Suni(Suni* Suni_main, string S_chuoioix) {
    (*this).S_chuoioix = S_chuoioix;
    this->Suni_main = Suni_main;
    if (this->Suni_main)
        Suni_main->Suni_luutru.push_back(this);
    else
        Virt_obj->Suni_luutru.push_back(this);
}

Suni* Suni::Virt_obj = new Suni();

// Метод переименования имени объекта
bool Suni::GanGiaTri(string S_thamsomoi) {
    if (this->Suni_main)
        for (int i = 0; i < Suni_main->Suni_luutru.size(); i++)
            if (Suni_main->Suni_luutru[i]->S_lay_ten() == S_thamsomoi)
                return false;

    S_chuoioix = S_thamsomoi;
    return true;
}

// Метод возврата адреса подчинённого объекта по имени
Suni* Suni::Doi_tuong_duoi(string S_thamsomoi2) {
    for (int j = 0; j < Suni_luutru.size(); j++)
        if (Suni_luutru[j]->S_lay_ten() == S_thamsomoi2)
            return Suni_luutru[j];

    return nullptr;
}

// Метод возврата указателя головного объекта
Suni* Suni::Doi_tuong_boss() {
    return Suni_main;
}

// Метод вывода дерева иерархии
void Suni::pearnet() {
    if (Doi_tuong_boss())
        cout << endl;

    int tabulia = 0;
    Suni* current = this;
    while (current->Doi_tuong_boss()) {
```

```

        current = current->Doi_tuong_boss();
        tabulia++;
    }
    for (int i = 0; i < tabulia; i++)
        cout << " ";
    cout << S_lay_ten();

    for (auto Suni_capduoi : Suni_luutru)
        Suni_capduoi->pearnet();
}

// Метод возврата значения свойства S_chuooix
string Suni::S_lay_ten() {
    return S_chuooix;
}

// Метод поиска объекта по имени в поддереве если уникальность объекта
имеется, иначе возврат nullptr
Suni* Suni::Timf_object_from_current(string S_thamso) {
    queue <Suni*> danhsach;
    Suni* Suni_tim = nullptr;

    danhsach.push(this);

    while (!danhsach.empty()) {
        if (danhsach.front()->S_lay_ten() == S_thamso) {
            if (!Suni_tim)
                Suni_tim = danhsach.front();
            else
                return nullptr;
        }

        for (auto Suni_capduoi : danhsach.front()->Suni_luutru)
            danhsach.push(Suni_capduoi);

        danhsach.pop();
    }
    return Suni_tim;
}

// Метод получения указателя на любой объект в составе дерева иерархии
объектов согласно пути
Suni* Suni::Timf_object_by_coord(string S_diachi) {
    string S_ten;
    int slash_index;
    Suni* Suni_doituong = nullptr;

    if (S_diachi.empty())
        return nullptr;

    if (S_diachi == "/") {
        Suni* quet = this;
        while (quet->Doi_tuong_boss())
            quet = quet->Doi_tuong_boss();
    }

```

```

        return quet;
    }

    if (S_diachi == ".")
        return this;

    if (S_diachi[0] == '/' && S_diachi[1] == '/') {
        S_ten = S_diachi.substr(2);
        return this->Timf_object_from_root(S_ten);
    }

    if (S_diachi[0] == '.') {
        S_ten = S_diachi.substr(1);
        return this->Timf_object_from_current(S_ten);
    }

    slash_index = S_diachi.find("/", 1);

    if (S_diachi[0] == '/') {
        if (slash_index != -1) {
            S_ten = S_diachi.substr(1, slash_index - 1);

            Suni_doituong = this->Timf_object_from_root(S_ten);
            if (Suni_doituong)
                return Suni_doituong-
>Timf_object_by_coord(S_diachi.substr(slash_index + 1));
            else
                return Suni_doituong;
        }
        else {
            S_ten = S_diachi.substr(1);
            return this->Timf_object_from_root(S_ten);
        }
    }
    else {
        if (slash_index != -1) {
            S_ten = S_diachi.substr(0, slash_index);

            Suni_doituong = this->Doi_tuong_duoi(S_ten);
            if (Suni_doituong)
                return Suni_doituong-
>Timf_object_by_coord(S_diachi.substr(slash_index + 1));
            else
                return Suni_doituong;
        }
        else {
            S_ten = S_diachi;
            return this->Doi_tuong_duoi(S_ten);
        }
    }
}

// Метод поиска подчинённого объекта корневого объекта по имени
Suni* Suni::Timf_object_from_root(string S_thamso) {
    Suni* quet = this;

```

```

    while (quet->Doi_tuong_boss())
        quet = quet->Doi_tuong_boss();
    return quet->Doi_tuong_duoi(S_thamso);
}

// Метод удаления объекта по имени
void Suni::xoa(string S_thamso) {
    Suni* current = Doi_tuong_duoi(S_thamso);

    if (current) {
        for (int i = 0; i < Suni_luutru.size(); i++) {
            if (Suni_luutru[i] == current) {
                Suni_luutru.erase(Suni_luutru.begin() + i);
                delete current;
                break;
            }
        }
    }
}

// Метод переопределения головного объекта для текущего в дереве иерархии
bool Suni::dichuyen(Suni* newHead) {
    if (this->Doi_tuong_boss() == newHead) {
        return true;
    }

    if (!(Doi_tuong_boss())) {
        return false;
    }

    Suni* quet = newHead;
    while (quet->Doi_tuong_boss()) {
        if (quet == this)
            return false;
        quet = quet->Doi_tuong_boss();
    }

    vector<Suni*>& v_thamchieu = Suni_main->Suni_luutru;
    for (int i = 0; i < v_thamchieu.size(); ++i) {
        if (v_thamchieu[i] == this) {
            v_thamchieu.erase(v_thamchieu.begin() + i);
            Suni_main = newHead;
            newHead->Suni_luutru.push_back(this);
            return true;
        }
    }
    return false;
}

// Метод установки готовности объекта
void Suni::Gan_trangthai(int thamso_trangthai) {
    if (thamso_trangthai) {
        Suni* quet = Suni_main;
        while (quet) {
            if (!(quet->trangthai))

```

```

        return;
        quet = quet->Doi_tuong_boss();
    }
    trangthai = thamso_trangthai;
}
else {
    for (auto Suni_capduoi : Suni_luutru) {
        Suni_capduoi->Gan_trangthai(thamso_trangthai);
    }
    trangthai = thamso_trangthai;
}
}

// Метод вывода класса базового и его пользовательского класса с их
// готовностью
void Suni::pearnet1() {
    if (Doi_tuong_boss())
        cout << endl;
    int tabulia = 0;
    Suni* current = this;
    while (current->Doi_tuong_boss()) {
        current = current->Doi_tuong_boss();
        tabulia++;
    }
    for (int i = 0; i < tabulia; i++)
        cout << "    ";
    cout << S_lay_ten();
    if (trangthai) {
        cout << " is ready";
    }
    else {
        cout << " is not ready";
    }
    for (auto Suni_capduoi : Suni_luutru)
        Suni_capduoi->pearnet1();
}

// Метод определения абсолютной пути до текущего объекта
string Suni::Toa_do() {
    Suni* quet = this;
    string S_ten = "/" + quet->S_chuooix;

    while (quet->Suni_main)
        quet = quet->Suni_main;
    if (quet == this) {
        S_ten = "/";
    }
    else {
        quet = this;
        while (quet->Suni_main->Suni_main) {
            quet = quet->Suni_main;
            S_ten = "/" + quet->S_chuooix + S_ten;
        }
    }
}

```

```

    return S_ten;
}

// Метод установки связи
void Suni::caidat_ketnoi(TYPE_SIGNAL ptr_tin_hieu,
    Suni* ptr_Suni,
    TYPE_HANDLER ptr_nhan_th) {
    ketnoi* thuc thi_ketnoi;

    //-----
    // Цикл для исключения повторного установления связи
    for (int i = 0; i < Suni_ddeens.size(); i++) {
        if (Suni_ddeens[i]->ptr_tin_hieu == ptr_tin_hieu &&
            Suni_ddeens[i]->ptr_Suni == ptr_Suni &&
            Suni_ddeens[i]->ptr_nhan_th == ptr_nhan_th) {
            return;
        }
    }
    thuc thi_ketnoi = new ketnoi(); // Создание объекта структуры
    // для хранения информации о новой связи

    thuc thi_ketnoi->ptr_tin_hieu = ptr_tin_hieu;
    thuc thi_ketnoi->ptr_Suni = ptr_Suni;
    thuc thi_ketnoi->ptr_nhan_th = ptr_nhan_th;
    Suni_ddeens.push_back(thuc thi_ketnoi); // Добавление новой связи
}

// Метод удаления связи
void Suni::xoa_ketnoi(TYPE_SIGNAL ptr_tin_hieu,
    Suni* ptr_Suni,
    TYPE_HANDLER ptr_nhan_th) {
    auto itr = Suni_ddeens.begin();

    while (itr != Suni_ddeens.end()) {
        if ((*itr)->ptr_tin_hieu == ptr_tin_hieu &&
            (*itr)->ptr_Suni == ptr_Suni &&
            (*itr)->ptr_nhan_th == ptr_nhan_th) {

            delete (*itr);
            itr = Suni_ddeens.erase(itr);

        }
        else {
            ++itr;
        }
    }
}

// Метод выдачи сигнала
void Suni::phat_tin_hieu(TYPE_SIGNAL ptr_tin_hieu, string tin_nhan, Suni*
S_ptr) {
    TYPE_HANDLER ptr_nhan_th;
    Suni* ptr_Suni;

    if (!(this->trangthai))

```



```

        return;

        (this->*ptr_tin_hieu)(tin_nhan);                // вызов метода
сигнала

        for (int i = 0; i < Suni_ddeens.size(); i++) {    // цикл по всем
обработчикам
            if (Suni_ddeens[i]->ptr_tin_hieu == ptr_tin_hieu    // определение
допустимого обработчика
                && Suni_ddeens[i]->ptr_Suni->trangthai
                && Suni_ddeens[i]->ptr_Suni == S_ptr) {

                ptr_nhan_th = Suni_ddeens[i]->ptr_nhan_th;
                ptr_Suni = Suni_ddeens[i]->ptr_Suni;

                (ptr_Suni->*ptr_nhan_th)(tin_nhan);        // вызов метода
обработчика
            }
        }
    }
}

// Метод возврата номера класса
int Suni::Lay_solop() {
    return 1;
}

// Метод приведения всех объектов в состоянии ГОТОВНОСТИ
void Suni::Active() {
    Gan_trangthai(1);

    for (int i = 0; i < Suni_luutru.size(); i++) {
        Suni_luutru[i]->Active();
    }
}

// Деструктор класса Suni
Suni::~~Suni() {
    for (int a = 0; a < Suni_luutru.size(); a++)
        delete Suni_luutru[a];
}

```

5.17 Файл Suni.h

Листинг 17 – Suni.h

```

#ifndef __SUNI__H
#define __SUNI__H

#include <iostream>
#include <vector>

```

```

#include <string>
#include <queue>

using namespace std;

class Suni;

#define SIGNAL_D(chuyen_tin_hieu) (TYPE_SIGNAL)(ampchuyen_tin_hieu)
#define HANDLER_D(nhan_tin_hieu) (TYPE_HANDLER)(ampnhan_tin_hieu)

typedef void (Suni :: *TYPE_SIGNAL)(string&);
typedef void (Suni :: *TYPE_HANDLER)(string);

struct ketnoi {
    TYPE_SIGNAL ptr_tin_hieu; // Структура задания одной связи
    Suni* ptr_Suni; // Указатель на метод сигнала
    TYPE_HANDLER ptr_nhan_th; // Указатель на целевой объект
    // Указатель на метод обработчика
};

class Suni {
protected:
    string S_chuooix;
    Suni* Suni_main;
    vector <Suni*> Suni_luutru;
    vector <ketnoi*> Suni_ddeens;
    int trangthai = 0;
    vector <string> Ncoffee;
    vector <int> Pcoffee;
    int sum = 0;
    int c5, c10;
    static Suni* Virt_obj;
public:
    Suni();
    Suni(Suni* Suni_main, string S_chuooix = "System");
    bool GanGiaTri(string S_thamsomoi);
    string S_lay_ten();
    Suni* Doi_tuong_duoi(string S_thamsomoi2);
    Suni* Doi_tuong_boss();
    void pearnet();
    void Gan_trangthai(int thamso_trangthai);
    void pearnet1();

    Suni* Timf_object_from_current(string S_thamso);
    Suni* Timf_object_from_root(string S_thamso);

    Suni* Timf_object_by_coord(string S_diachi);

    bool dichuyen(Suni* newHead);
    void xoa(string S_thamso);

    string Toa_do();
    void caidat_ketnoi(TYPE_SIGNAL ptr_tin_hieu,
        Suni* ptr_Suni,
        TYPE_HANDLER ptr_nhan_th );

```

```

        void xoa_ketnoi(TYPE_SIGNAL ptr_tin_hieu,
                        Suni* ptr_Suni,
                        TYPE_HANDLER ptr_nhan_th );

        void phat_tin_hieu(TYPE_SIGNAL ptr_tin_hieu, string tin_nhan, Suni*
S_ptr);

        virtual int Lay_solop();
        void Active();

        friend class Suni2;
        friend class Suni3;
        friend class Suni4;

        ~Suni();
};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 21.

Таблица 21 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
3 Espresso Americano Cappuchino 25 50 50 3 5 50 Coffee Cappuchino 10 10 10 Coffee Espresso 5 5 Refund money 5 100 Cancel	Ready to work The amount: 50 Take the coffee Cappuchino Ready to work The amount: 10 The amount: 20 The amount: 30 Take the coffee Espresso Take the change: 10 * 0 rub., 5 * 1 rub. Ready to work The amount: 5 The amount: 10 Take the money: 10 * 1 rub., 5 * 0 rub. Ready to work The amount: 5 Take the money back, no change Turned off	Ready to work The amount: 50 Take the coffee Cappuchino Ready to work The amount: 10 The amount: 20 The amount: 30 Take the coffee Espresso Take the change: 10 * 0 rub., 5 * 1 rub. Ready to work The amount: 5 The amount: 10 Take the money: 10 * 1 rub., 5 * 0 rub. Ready to work The amount: 5 Take the money back, no change Turned off
3 S A P 50 50 50 11 11 50 Coffee S 100 Coffee A 10 10 Coffee P 100 Coffee A 100 Coffee A SHOWTREE	Ready to work The amount: 50 Take the coffee S Ready to work The amount: 100 Take the coffee A Take the change: 10 * 5 rub., 5 * 0 rub. Ready to work The amount: 10 The amount: 20 The amount: 30 There is not enough money The amount: 130 Take the coffee A Take the change: 10 * 8 rub., 5 * 0 rub.	Ready to work The amount: 50 Take the coffee S Ready to work The amount: 100 Take the coffee A Take the change: 10 * 5 rub., 5 * 0 rub. Ready to work The amount: 10 The amount: 20 The amount: 30 There is not enough money The amount: 130 Take the coffee A Take the change: 10 * 8 rub., 5 * 0 rub.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	<p>Ready to work Take the money back, no change There is not enough money System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready</p>	<p>Ready to work Take the money back, no change There is not enough money System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready</p>
<p>3 Espresso Americano Cappuccino 25 50 50 3 5 10 Coffee Cappuccino 50 Coffee Espresso SHOWTREE</p>	<p>Ready to work The amount: 10 There is not enough money The amount: 60 Take the coffee Espresso Take the change: 10 * 3 rub., 5 * 1 rub. Ready to work System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready</p>	<p>Ready to work The amount: 10 There is not enough money The amount: 60 Take the coffee Espresso Take the change: 10 * 3 rub., 5 * 1 rub. Ready to work System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready</p>
<p>3 Espresso Americano Cappuccino 25 50 50 3 5 50 Coffee Cappuccino 10 10 10 Coffee Espresso</p>	<p>Ready to work The amount: 50 Take the coffee Cappuccino Ready to work The amount: 10 The amount: 20 The amount: 30 Take the coffee Espresso Take the change: 10</p>	<p>Ready to work The amount: 50 Take the coffee Cappuccino Ready to work The amount: 10 The amount: 20 The amount: 30 Take the coffee Espresso Take the change: 10</p>

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 5 Refund money 5 100 Cancel	* 0 rub., 5 * 1 rub. Ready to work The amount: 5 The amount: 10 Take the money: 10 * 1 rub., 5 * 0 rub. Ready to work The amount: 5 Take the money back, no change Turned off	* 0 rub., 5 * 1 rub. Ready to work The amount: 5 The amount: 10 Take the money: 10 * 1 rub., 5 * 0 rub. Ready to work The amount: 5 Take the money back, no change Turned off
3 Espresso Americano Cappuccino 25 50 50 3 5 100 50 Refund money 10 10 5 Coffee Espresso SHOWTREE	Ready to work Take the money back, no change The amount: 50 Take the money: 10 * 5 rub., 5 * 0 rub. Ready to work The amount: 10 The amount: 20 The amount: 25 Take the coffee Espresso Ready to work System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready	Ready to work Take the money back, no change The amount: 50 Take the money: 10 * 5 rub., 5 * 0 rub. Ready to work The amount: 10 The amount: 20 The amount: 25 Take the coffee Espresso Ready to work System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready
3 Espresso Americano Cappuccino 25 50 50 3 5 50 Coffee Espresso Cancel	Ready to work The amount: 50 Take the coffee Espresso Take the change: 10 * 2 rub., 5 * 1 rub. Ready to work Turned off	Ready to work The amount: 50 Take the coffee Espresso Take the change: 10 * 2 rub., 5 * 1 rub. Ready to work Turned off
1 Espresso 25 1 1	Ready to work System is ready Command Reader is ready	Ready to work System is ready Command Reader

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
SHOWTREE	Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready	is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready
1 Espresso 25 0 10 100 SHOWTREE	Ready to work The amount: 100 System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready	Ready to work The amount: 100 System is ready Command Reader is ready Control Panel is ready Money Receiver is ready Change Returner is ready Display is ready Coffee Dispenser is ready
3 Suni Cà_phê_phin Cà_phê_sữa_đá 0 0 0 5 5 50 Coffee Suni 100 Coffee Cà_phê_sữa_đá 10 10 10 Coffee Cà_phê_phin SHOWTREE	Ready to work The amount: 50 Take the coffee Suni Take the change: 10 * 5 rub., 5 * 0 rub. Ready to work Take the money back, no change Take the coffee Cà_phê_sữa_đá Ready to work The amount: 10 The amount: 20 The amount: 30 Take the coffee Cà_phê_phin Take the change: 10 * 3 rub., 5 * 0 rub. Ready to work System is ready Command Reader is ready Control Panel is ready Money Receiver	Ready to work The amount: 50 Take the coffee Suni Take the change: 10 * 5 rub., 5 * 0 rub. Ready to work Take the money back, no change Take the coffee Cà_phê_sữa_đá Ready to work The amount: 10 The amount: 20 The amount: 30 Take the coffee Cà_phê_phin Take the change: 10 * 3 rub., 5 * 0 rub. Ready to work System is ready Command Reader is ready Control Panel is ready Money Receiver

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	is ready Change Returner is ready Display is ready Coffee Dispenser is ready	is ready Change Returner is ready Display is ready Coffee Dispenser is ready

ЗАКЛЮЧЕНИЕ

Благодаря курсовым вариантам и упражнениям мы узнаем **преимущества** ООП:

- **Инкапсуляция:** ООП позволяет инкапсулировать свойства и методы, работающие с этими свойствами, в объект. Это помогает защитить данные от внешнего вмешательства, повышая безопасность и надежность программы.
- **Наследование:** ООП позволяет одному классу наследовать свойства и методы другого класса. Это способствует повторному использованию кода, снижению необходимости переписывания кода и созданию более управляемой иерархической структуры.
- **Полиморфизм:** ООП поддерживает полиморфизм, позволяя методу иметь несколько различных форм. Это помогает манипулировать объектами более гибко, облегчая расширение и сопровождение системы.

ООП помогает нам, предоставляя ясную и понятную структуру программирования, повышая повторное использование и уменьшая ошибки в процессе разработки программного обеспечения. Он также создаёт удобные условия для разработки и обслуживания исходного кода эффективным способом. Оно также помогает нам предотвратить негативные события, предоставляя механизмы безопасности, такие как сокрытие данных и ограничение доступа к частям исходного кода, обеспечивая защиту важной информации и предотвращая несанкционированный доступ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.
2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avvora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avvora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
4. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).