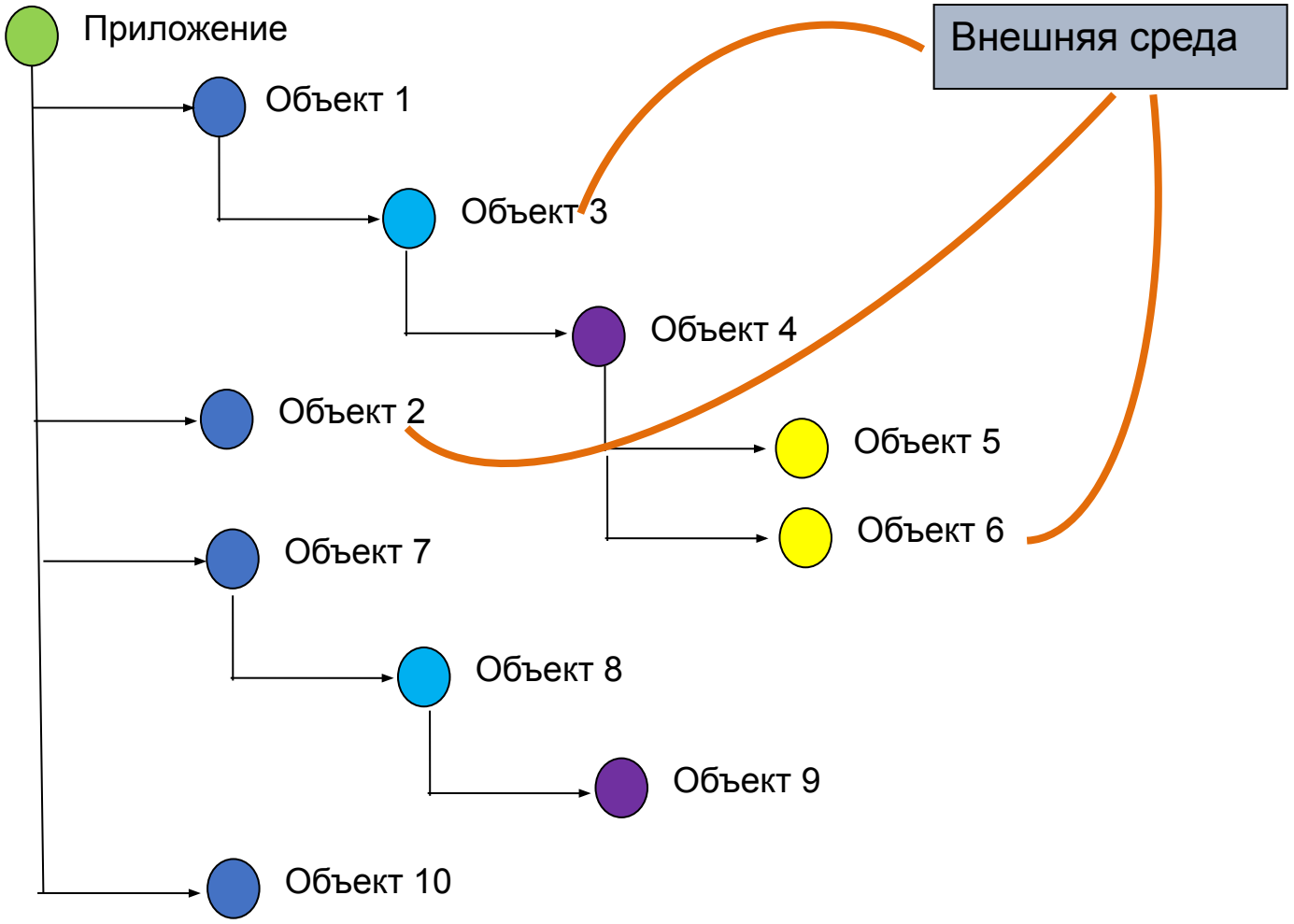
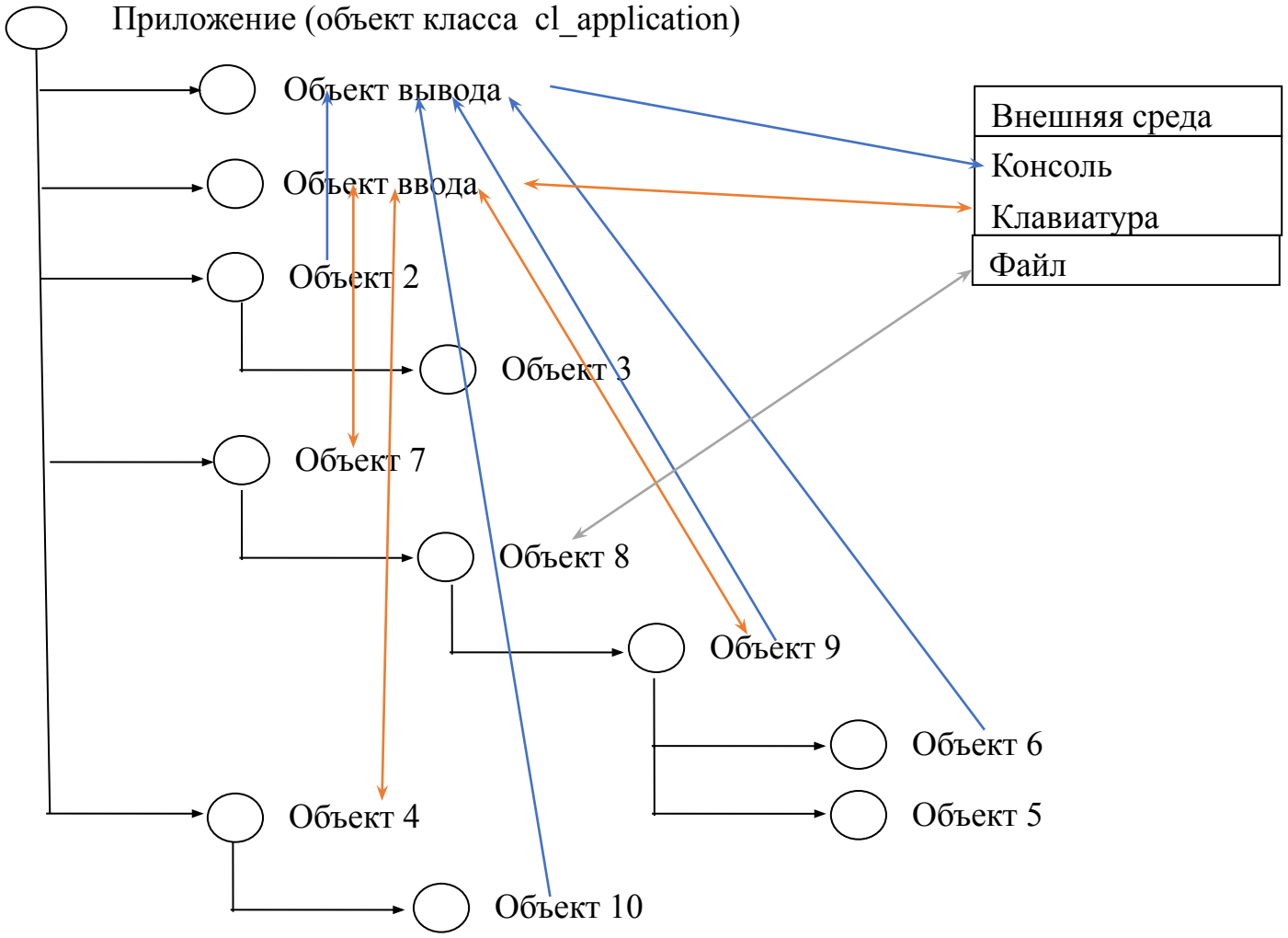
Объектно-ориентированное программирование на алгоритмическом языке C++

МИРЭА, Институт Информационных технологий, кафедра Вычислительной техники

Схема архитектуры программы
Дерево объектов



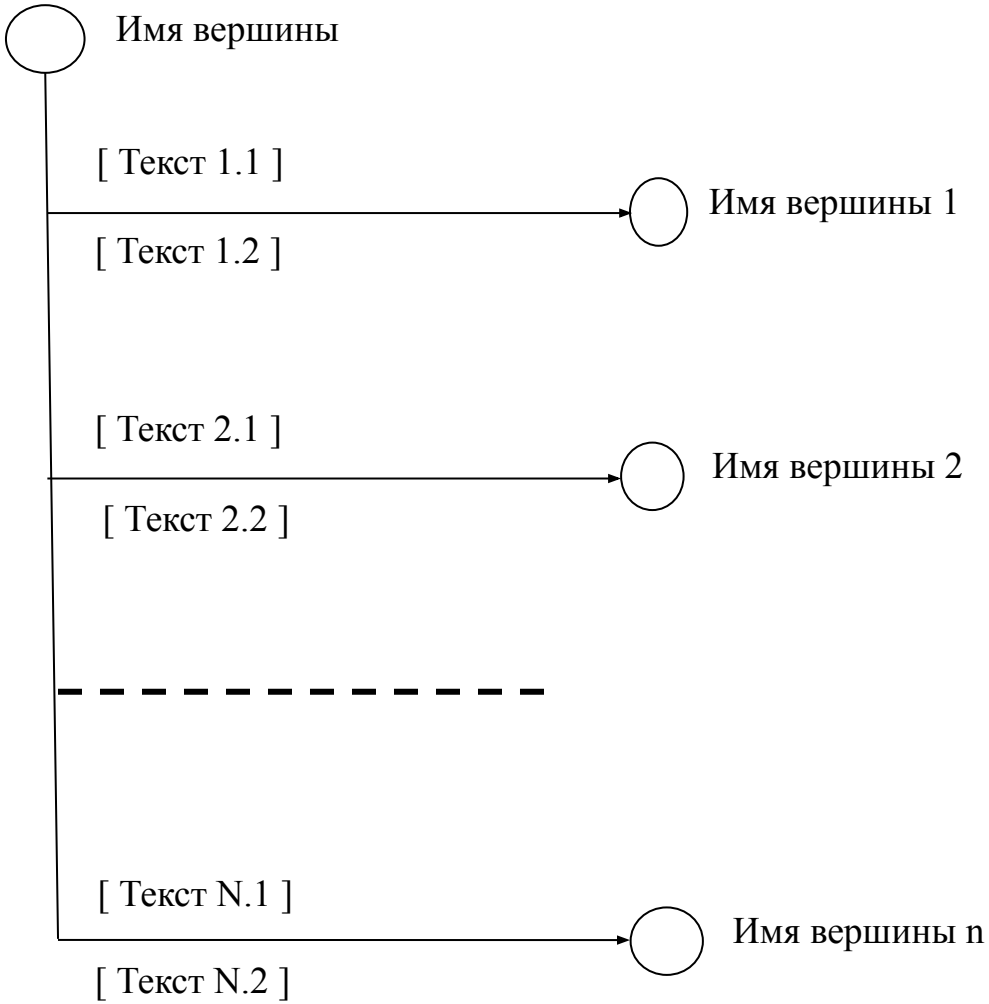
Детализация схемы архитектуры программы



Содержание этапов разработки программы

Процедурное программирование	Объектно-ориентированное программирование
Постановка задачи (что)	Постановка задачи (что)
Метод решения (чем)	Множество объектов. Методы решения. (чем)
Алгоритм решения задачи (как)	Архитектура программы-системы, иерархия объектов. Схема взаимодействия объектов. Алгоритм функционирования, решения задачи (как).
Блок-схема алгоритма	Схема иерархии наследования классов. Схема архитектуры программы. Схема взаимодействия объектов. Схема алгоритма решения задачи.
Код программы	Код описания классов. Код конструирования системы (архитектуры программы). Код взаимодействия объектов. Код алгоритма решения задачи.
Тестирование и отладка	Тестирование и отладка
Доработка документации	Доработка документации
Сдача программы и сопроводительной документации	Сдача программы и сопроводительной документации

Ориентированный нагруженный граф



Иерархия наследования классов

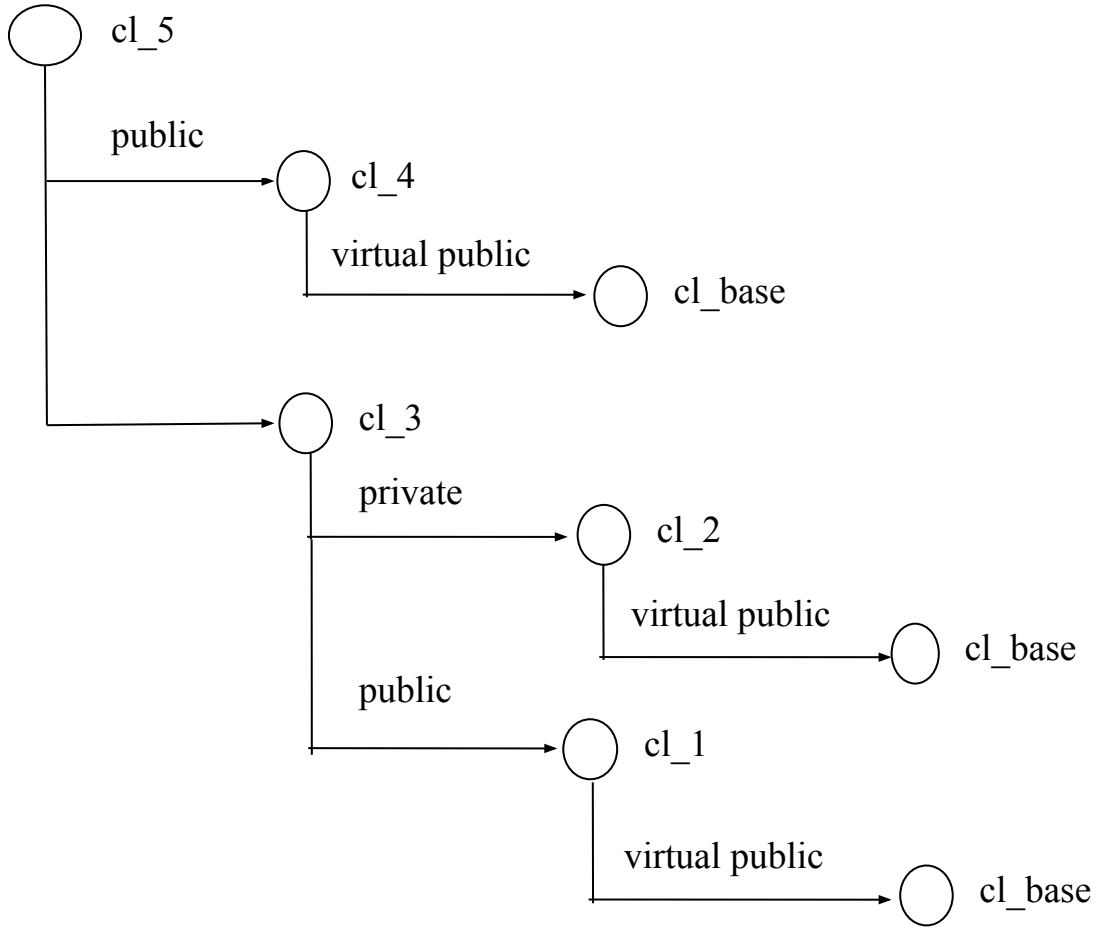
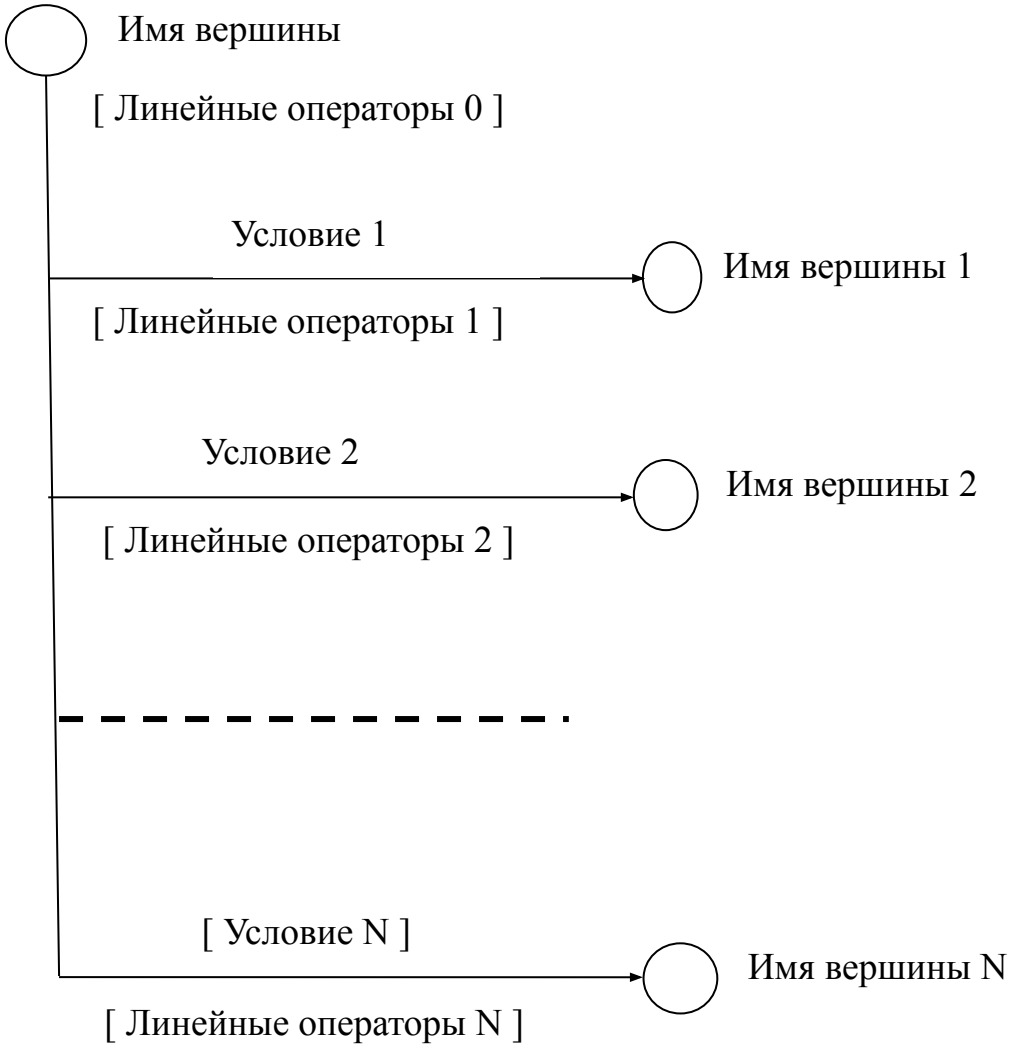


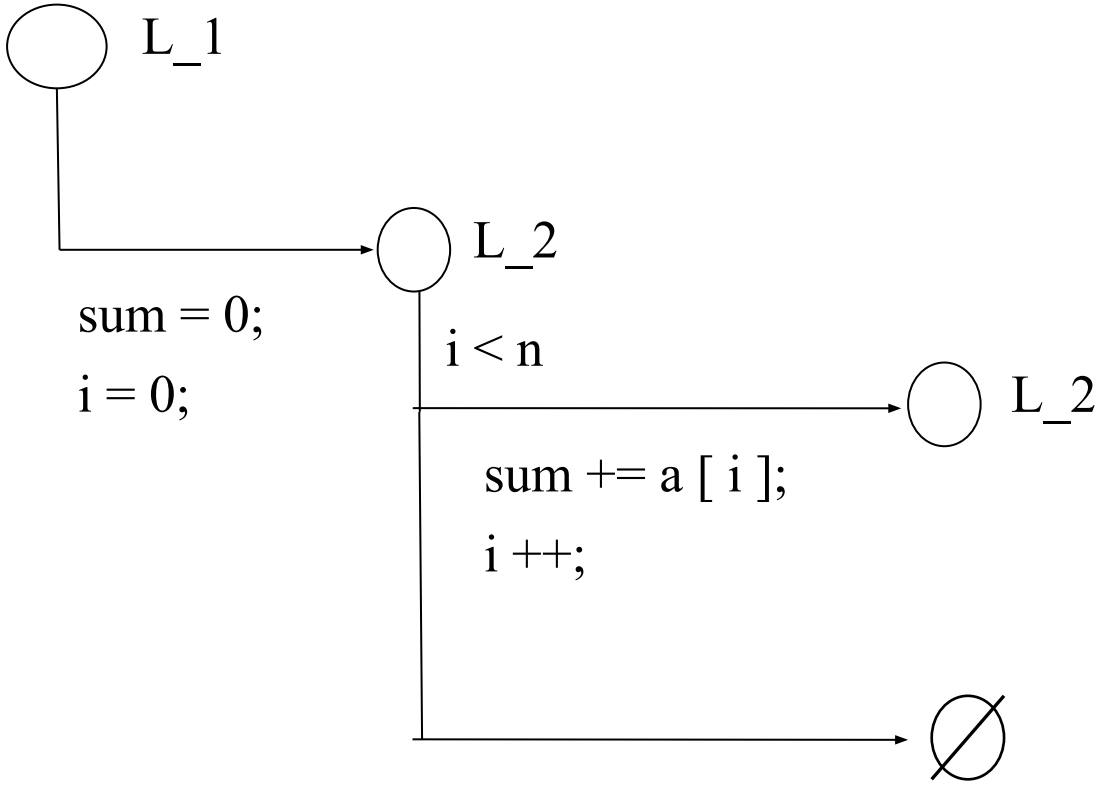
Схема алгоритма



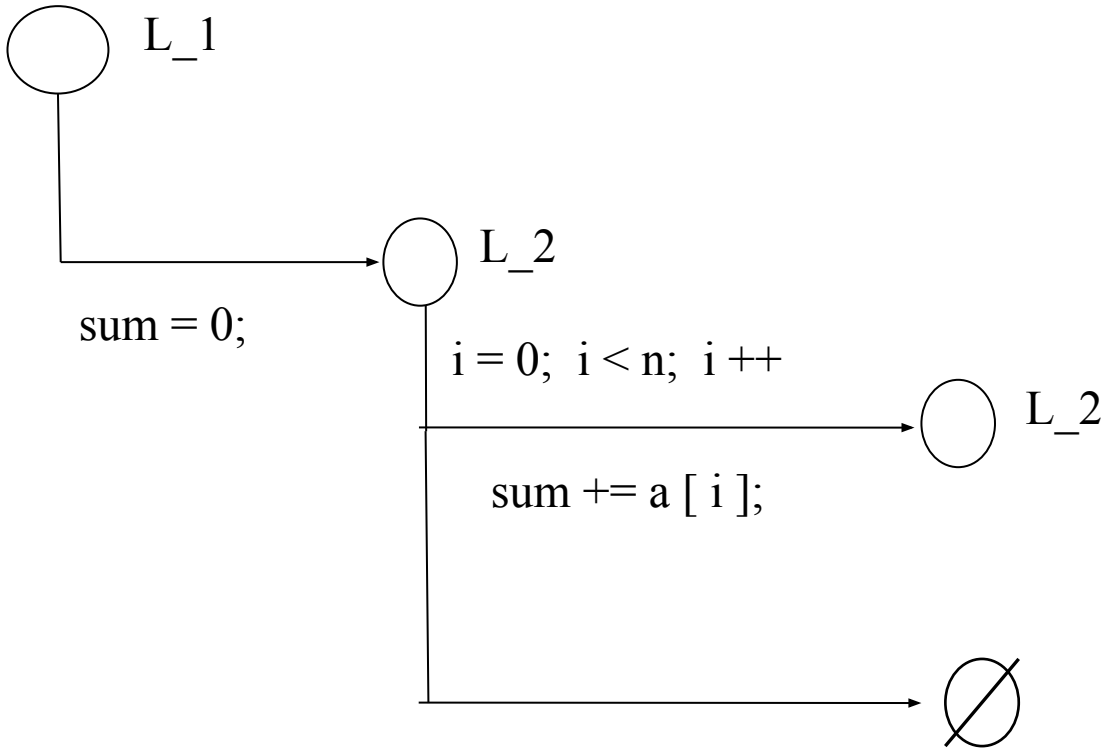
Табличное представление алгоритма

Вершина комплекса	**Предикат**	**Процедуры**	**Вершина перехода**
Имя вершины		Линейные операторы 0	
	Условие 1	Линейные операторы 1	**Имя вершины 1**
	Условие 2	Линейные операторы 2	**Имя вершины 2**
	…	…	…
	Условие N	**Линейные операторы N**	**Имя вершины N**

Пример представления алгоритма



Пример представления алгоритма



Достоинства данной графической нотации описания блок-схем алгоритма

- Простота используемых графических элементов.
- Графическое изображение алгоритма строится слева на право и сверху вниз.
- Взаимно однозначность реализации кода на языке C++.
- Возможность и взаимно однозначность представления алгоритма в табличном виде.

Жизненный цикл виртуального объекта

Реализация жизненного цикла виртуального объекта на языке C++

Описание	Класс
Создание	Отработка конструктора объекта. Выделение памяти и исходная инициализация.
Старт объекта	Завершение работы конструктора объекта. Начало функционирования.
Функционирование	Участие объекта в работе (в алгоритме) программы
Остановка	Вызов деструктора объекта
Демонтаж	Отработка деструктора объекта
Завершение	Завершение работы деструктора объекта. Освобождение выделенной объекту памяти.

Взаимодействие с объектом

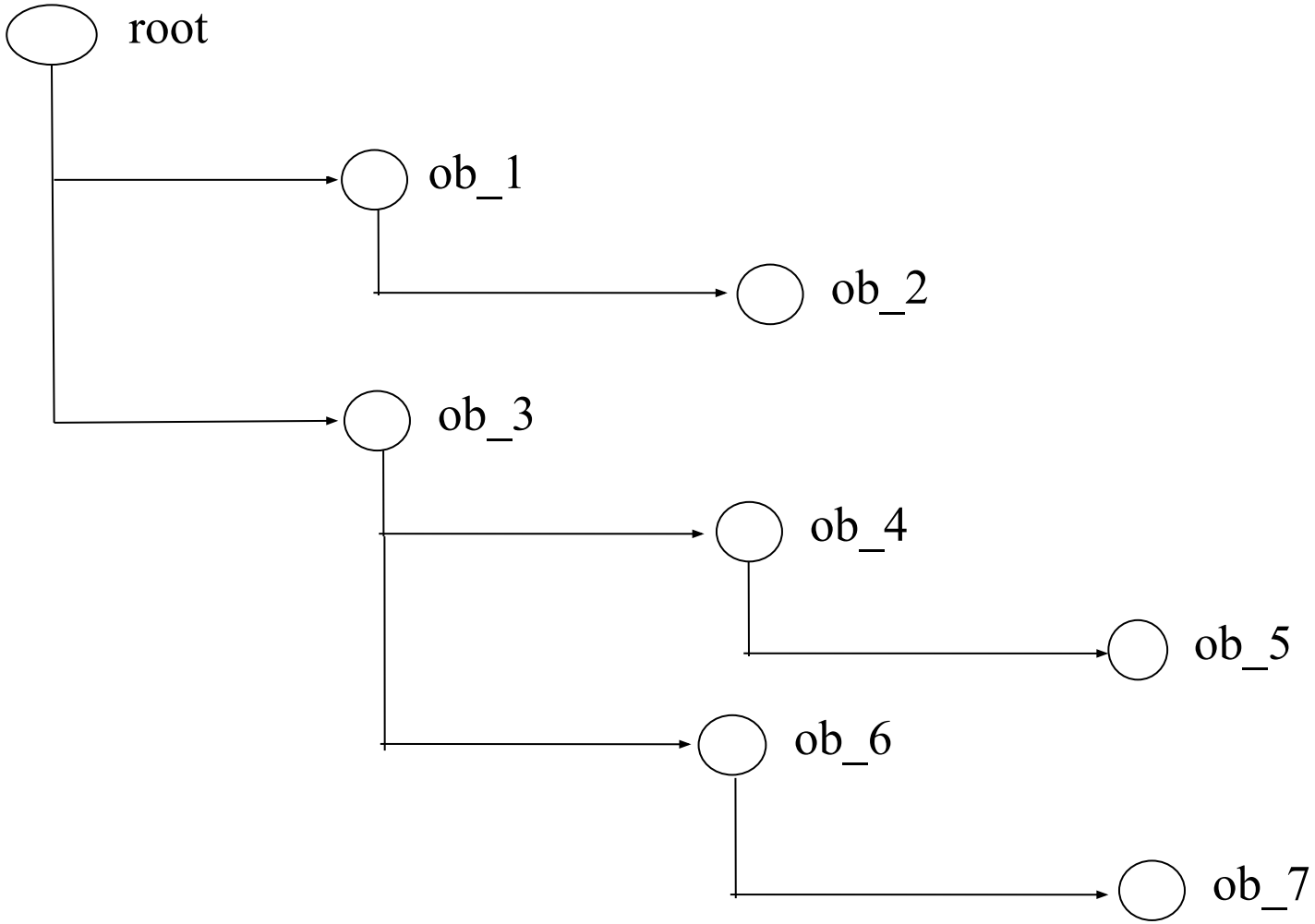
Обращение к объекту для выполнения определенного метода (реакции) инициируется по:

- Внешнему событию.
- Внутреннему событию.
- Сигналу.
- Непосредственным вызов метода объекта.

Элементы языка C++ для построения программ

Описание объекта	Описание класса
Создание объекта	Создание объекта на базе определенного класса посредством конструктора объекта
Построение конструкции объекта	Наследственность. Включение объектов других классов в описании свойств класса.
Взаимодействие объектов в рамках конструкции объекта	Непосредственный вызов метода объекта. Сигнал от объекта ⇒ обработчик объекта.
Архитектура программы-системы (конструктивное построение)	Построение иерархии объектов согласно схеме взаимосвязи объектов в рамках программы
Схема взаимодействия объектов в составе программы-системы	Построение интерфейсов между объектами программы согласно схеме взаимодействия. Сигнал от объекта ⇒ обработчик объекта.
Схема взаимодействия объектов программы внешней средой	Построение интерфейсов между объектами программы и внешней средой согласно схеме. Внешнее (системное) событие ⇒ обработчик объекта. Обращение к интерфейсу со стороны внешней среди, передача данных (ввод). Обращение к интерфейсу внешней среди со стороны объекта, передача данных (вывод).

Дерево иерархии объектов



/root/ob_3/ob_4/ob_5

Базовый класс

- свойства:
  - наименование объекта;
  - ссылку на головной объект на дереве иерархии объектов;
  - перечень ссылок на объектов-потомков.
- методы:
  - присвоить имя объекту;
  - получить имя объекта;
  - определить ссылку на головной объект;
  - добавить новый объект в перечне объектов-потомков;
  - получить ссылку на объект потомок по имени объекта;
  - удалить объект из перечня объектов-потомков.

Примерная заголовочная часть базового класса

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

class cl_base   {
public:
    cl_base ( cl_base * p_parent = 0 );

    void      set_object_name ( string    object_name );
    string    get_object_name ( );
    void      set_parent      ( cl_base * p_parent    );
    void      add_child       ( cl_base * p_child     );
    void      delete_child    ( string    object_name );
    cl_base * get_child       ( string    object_name );
    cl_base * get_object      ( string    object_path );

    vector < cl_base * > children;       // ссылки на потомков
    vector < cl_base * > :: iterator  it_child;
 private:
    string       object_name;            // наименование объекта
    cl_base    * p_parent;               // ссылка на головной объект
};
```

Класс приложение

Класс приложения базируется на класс базового объекта. Класс содержит:

методы:

- построить дерево иерархии объектов;
- запустить приложение;
- вывод дерева иерархии объектов на консоль.

Примерная заголовочная часть класса приложения

```
#include "cl_base.h"

class cl_application : public cl_base
{
public:
    cl_application ( );

    void bild_tree_objects ( );
    int  exec_app          ( );

    void show_object_tree  ( );

private:
    void show_object_next ( cl_base * ob_parent, int i_level );
};
```

Вид вывода на консоль дерева иерархии объектов

```
root
     ob_1
          ob_2
     ob_3
          ob_4
               ob_5
          ob_6
               ob_7
```

Основная функция main

Основная функция реализует стандартный алгоритм.

1. Создать объект приложение.
2. Построить дерево объектов.
3. Запустить приложение.

```
#include <iostream>
using namespace std;

#include "cl_application.h"

int main ( )
{
    cl_application  ob_application;

    ob_application.bild_tree_objects ( );

    return ob_application.exec_app   ( );
}
```

Лабораторные работы

1. Создание базового класса объекта.
2. Создание класса приложения.
3. Конструктивное построение приложения.
   - Вывод на консоль дерева иерархии объектов.
   - Вывод на консоль ветки дерева иерархии объектов.
   - Динамическое добавление объекта в дереве иерархии.
   - Динамическое удаление объекта из дерева иерархии с последующей веткой.
   - Копирование ветки дерева иерархии.
   - Вставка ветки дерева иерархии.
4. Координаты объекта на дереве иерархии, по аналогии XPath.
   - Метод получения ссылки на объект исходя из координаты (пути) объекта в дереве иерархии.
   - Метод получения ссылки на объект исходя из относительной координаты (пути) объекта в дереве иерархии.
5. Объекты ввода и вывода.
6. Простые сигналы между объектами.