

ГЛАВА 7.

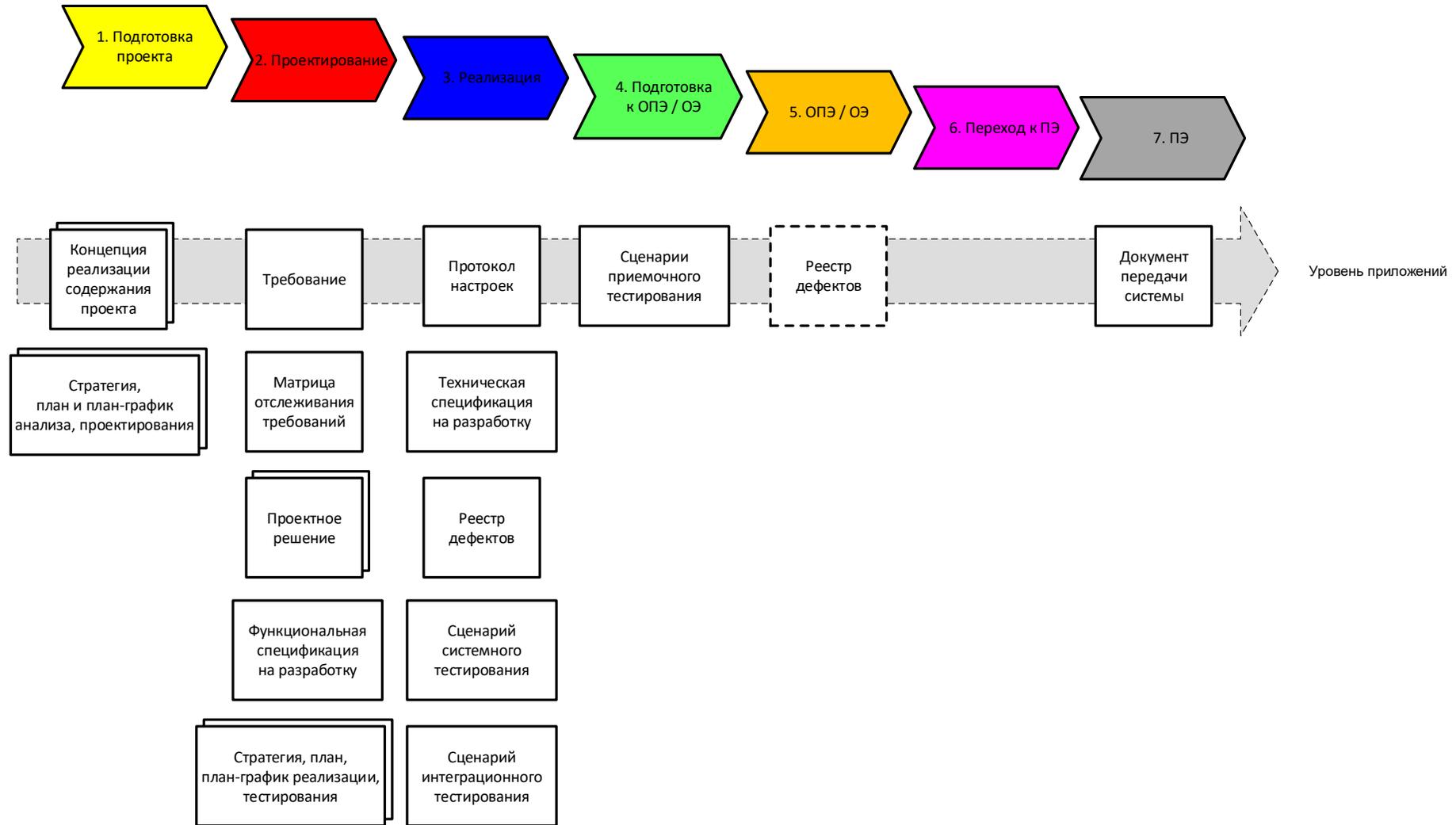
УРОВЕНЬ ПРИЛОЖЕНИЙ ПРИ ВНЕДРЕНИИ ОПТИКО- ИНФОРМАЦИОННЫХ СИСТЕМ

- Документирование уровня приложений
- Архитектура приложений предприятия
- Концепция анализа
- Области знаний для реализации программ
- Концепция реализации
- Концепция тестирования
- Трехуровневая структура описания программ

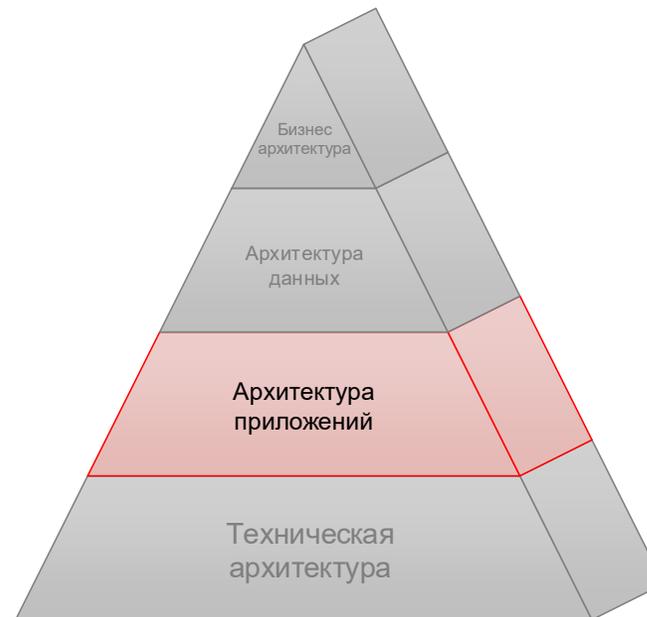
7.2. Разграничение ответственности по уровням



7.3. Документирование уровня приложений



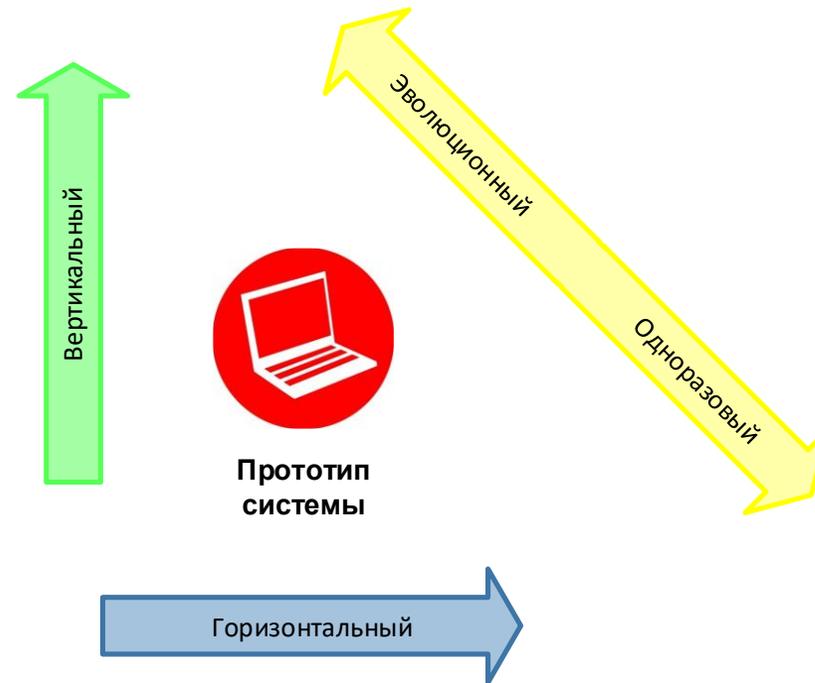
7.4. Архитектура приложений предприятия



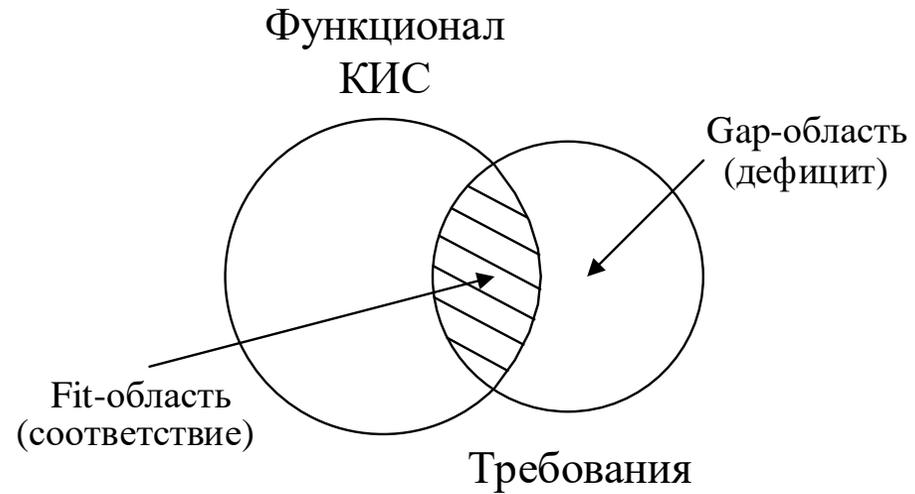
7.5. Способы анализа требований



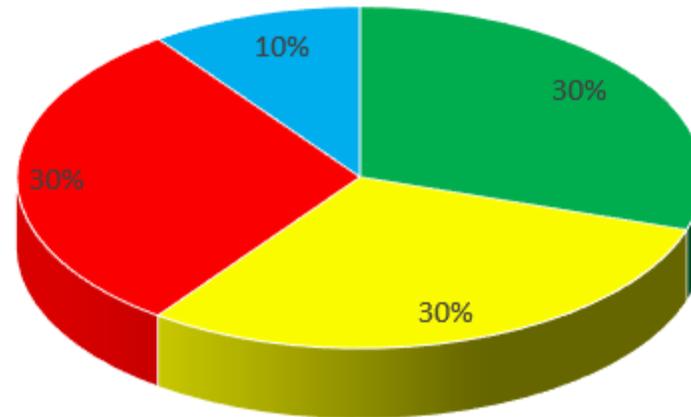
7.6. Прототипирование системы



7.7. Fit/Gap-анализ

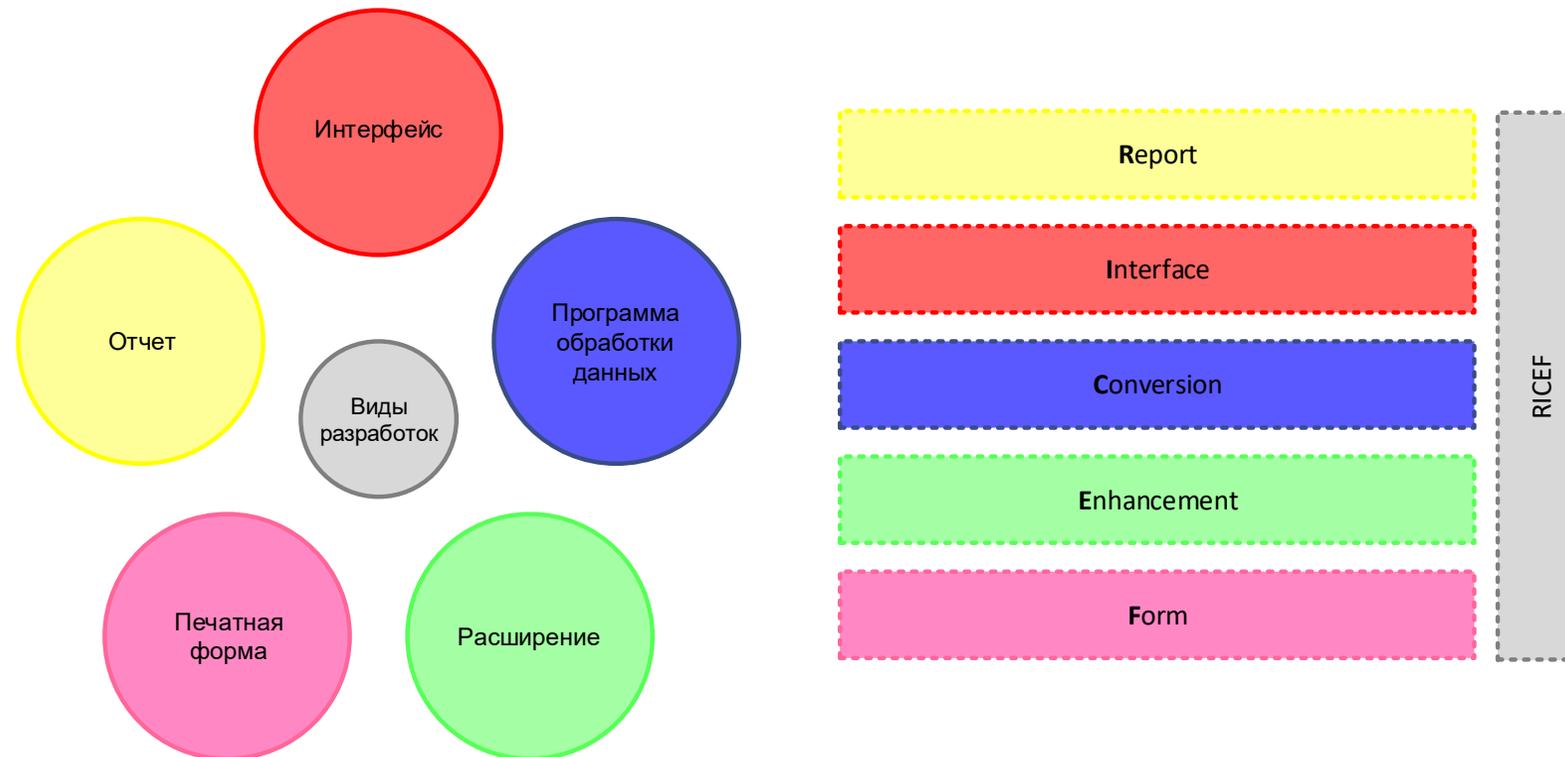


7.8. Статистика Fit/Gap-анализа



- Реализовано
- Требуется доработки
- Не реализовано
- Прочее

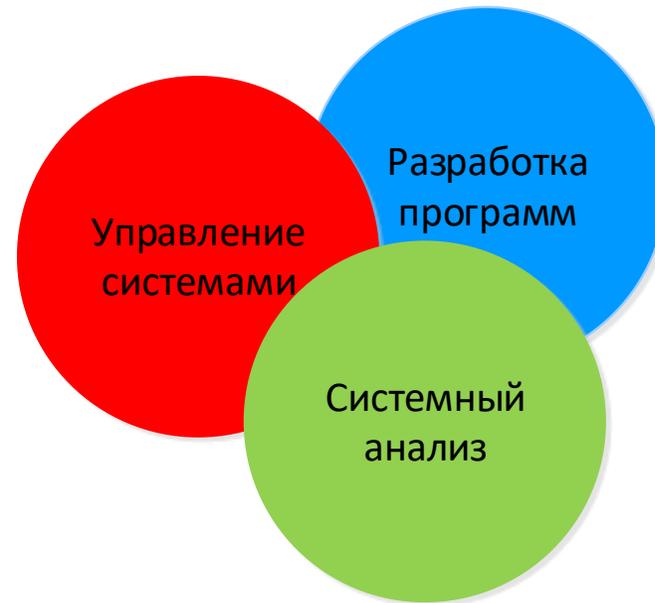
7.9. Виды разработок Gap-области



7.10. Матрица отслеживания требований

Требование	Fit/Gap	RICEF	Кол-во разработок	Проектное решение	Функциональная спецификация на разработку
Требование 1	Fit	–	–	MM_01	–
Требование 2	Gap	R	1	MM_01	MM_01.1
Требование 3	Fit	–	–	SD_02	–

7.11. Области знаний для реализации программ



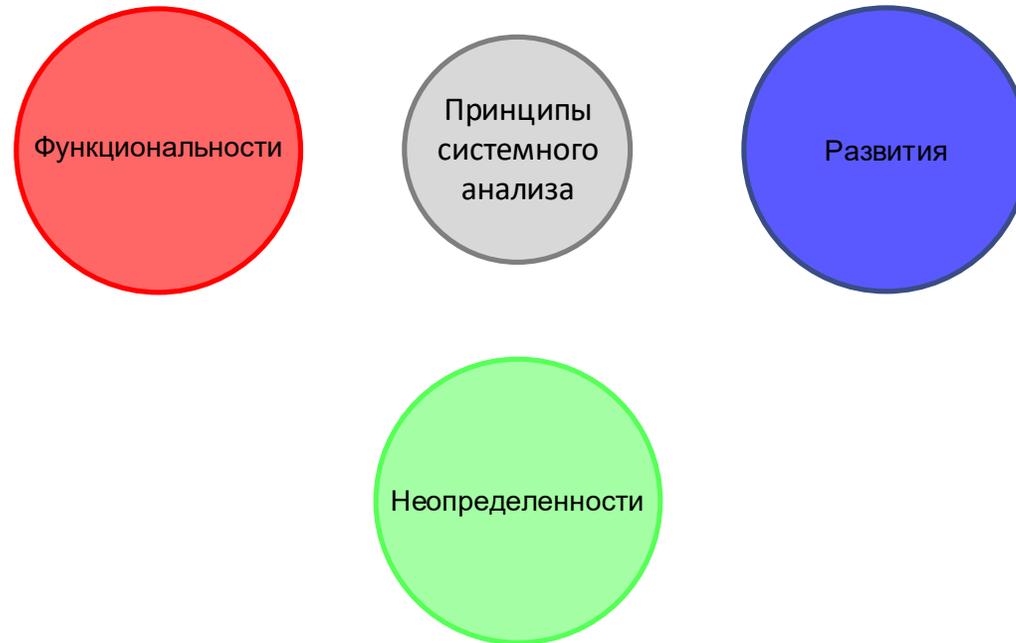
7.12. Принципы управления системами



7.13. Принципы разработки программ



7.14. Принципы системного анализа



7.15. Обобщение принципов разработки (1 из 3)

Вид разработки	Принцип	Рекомендация
Все разработки	Развития	Общий вид решения задачи
Все разработки	Функциональности	Корректное наименование программных разработок
Все разработки	Развития	Проверка полномочий при запуске программы
Все разработки	Контур обратной связи	Информирование пользователя о результатах работы программы
Все разработки	Модульности	Разработка и использование функций для часто повторяющихся операций
Все разработки	Развития	Вынесение константных переменных в отдельный модуль

7.15. Обобщение принципов разработки (2 из 3)

Вид разработки	Принцип	Рекомендация
Отчет	Функциональности	Отображение формата полей на селекционном экране
Отчет	Контур обратной связи	Ограничение выбора данных на основе полномочий
Отчет	Развития	Вывод максимального количества полей в таблицу отображения данных

Вид разработки	Принцип	Рекомендация
Интерфейс	Надежности	Передача и проверка сумма при выгрузки и загрузки
Интерфейс	Неопределенности	Блокировка повторной загрузки данных

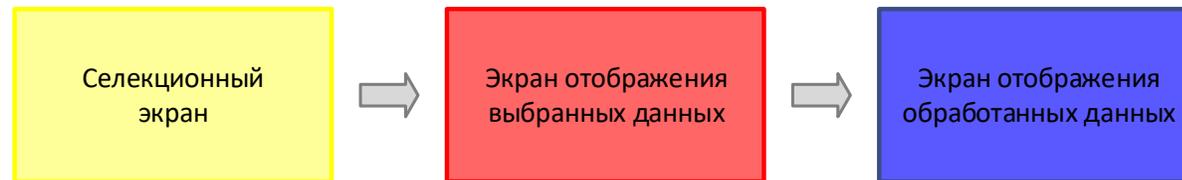
7.15. Обобщение принципов разработки (3 из 3)

Вид разработки	Принцип	Рекомендация
Программа обработки данных	Надежности	Пометка обработанных данных
Программа обработки данных	Надежности	Вывод стандартного описания системных ошибок в экран обработки
Программа обработки данных	Контур обратной связи	Хранение потока документов при обработке большого числа объектов

Вид разработки	Принцип	Рекомендация
Расширение	Неопределенности	Защита от дурака

Вид разработки	Принцип	Рекомендация
Печатная форма	«По умолчанию»	Выбор данных для заголовка из первой неудаленной позиции документа

7.16. Трехуровневая структура описания программ



7.17. Пример трехуровневой структуры

а) Экран выбора документов материалов

Год:

Документ материала:

Завод:

б) Экран отображения выбранных документов материалов

Статус	Год	Документ	Позиция	Краткий текст
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000000	10	
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000000	20	
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000001	30	Сертификат №121
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000001	40	
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000002	20	
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000003	10	
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000004	10	
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000005	20	№1, 2, 3
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	2014	5100000006	20	

Создать сертификат

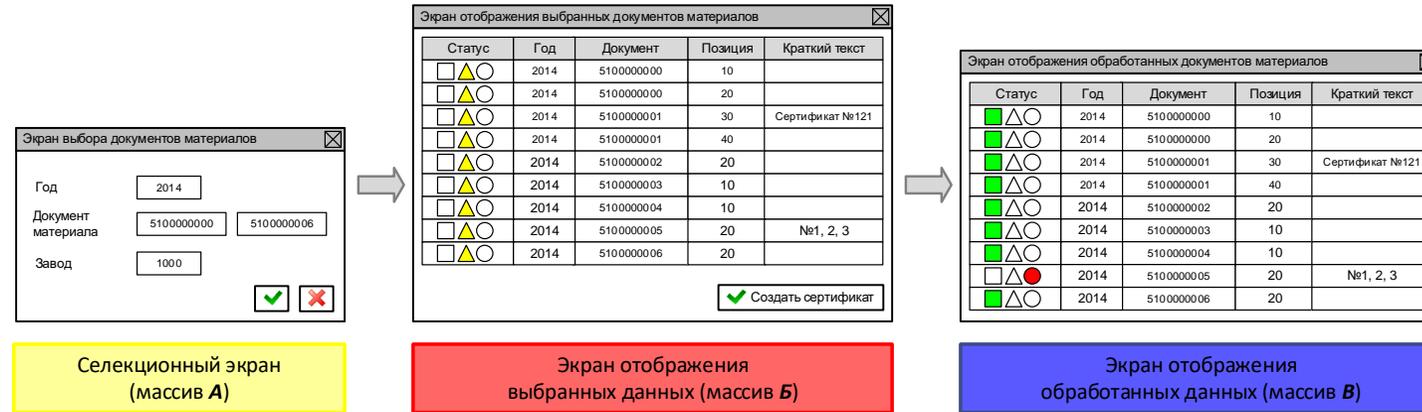
в) Экран отображения обработанных документов материалов

Статус	Год	Документ	Позиция	Краткий текст
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000000	10	
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000000	20	
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000001	30	Сертификат №121
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000001	40	
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000002	20	
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000003	10	
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000004	10	
<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	2014	5100000005	20	№1, 2, 3
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2014	5100000006	20	

7.18. Виды программ и трехуровневая структура

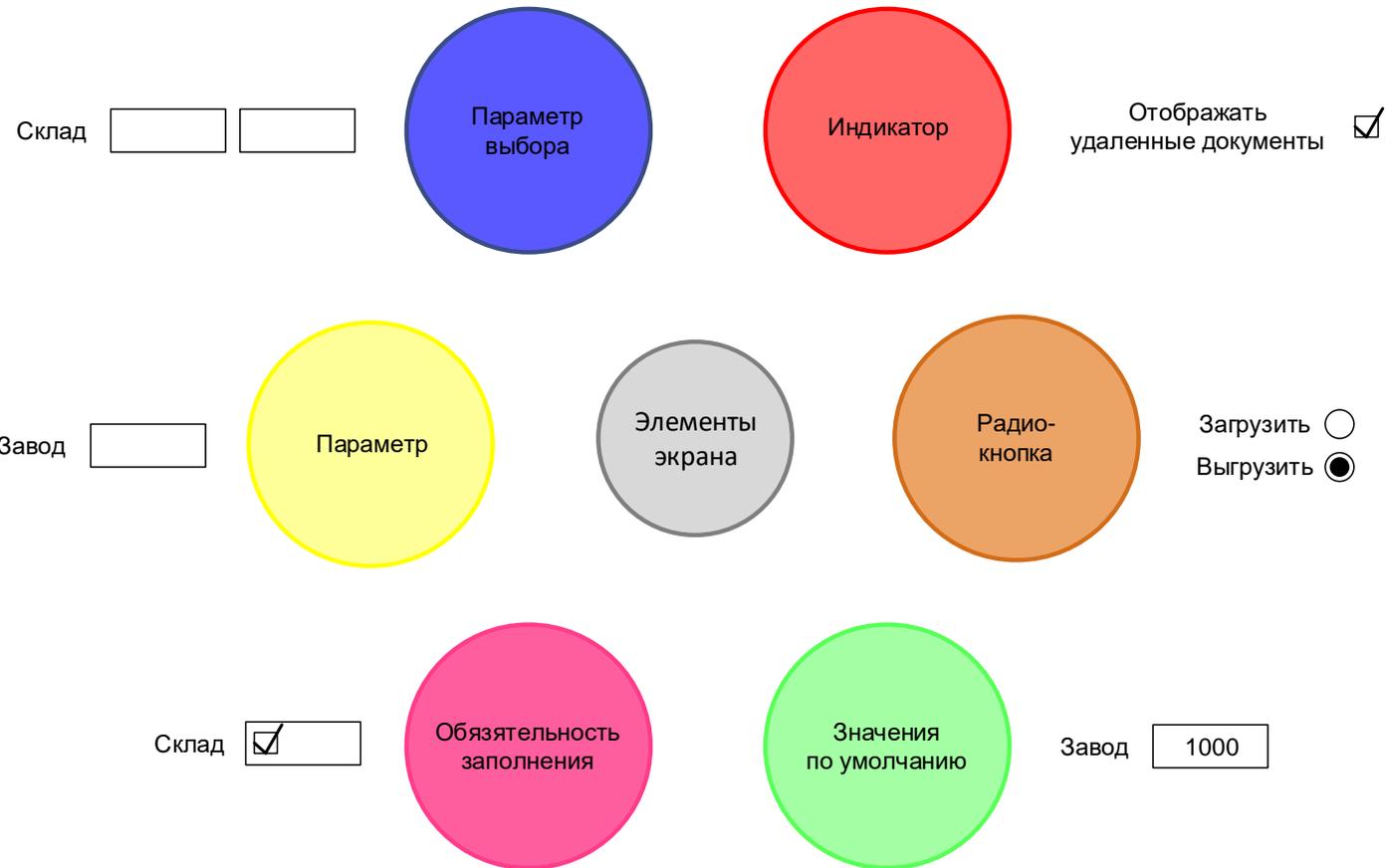
Вид разработки	Селекционный экран	Экран выбранных данных	Экран обработанных данных
Отчет	✓	✓	
Интерфейс	✓	✓	✓
Программа обработки данных	✓	✓	✓
Расширение			
Печатная форма	✓	✓	

7.19. Преобразование данных в трехуровневой структуре

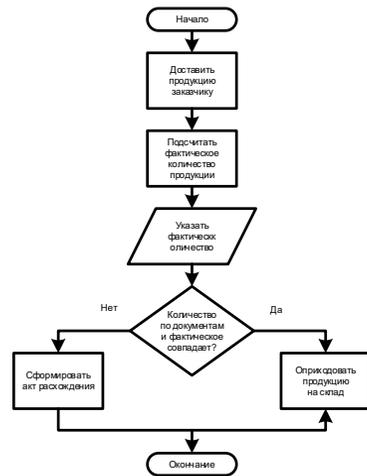


1. Определить все поля и кнопки селекционного экрана **A**
2. Определить все поля и кнопки экрана отображения выбранных данных **B**
3. Записать алгоритм заполнения полей 2 на основе данных из 1 **$B = f(A)$**
4. Определить все поля и кнопки экрана отображения обработанных данных **B**
5. Записать алгоритм заполнения полей 4 на основе данных из 2 **$B = f(B)$**

7.20. Элементы экранов программ



7.21. Описание алгоритмов заполнения полей



Блок-схема алгоритма

Способы описания алгоритмов

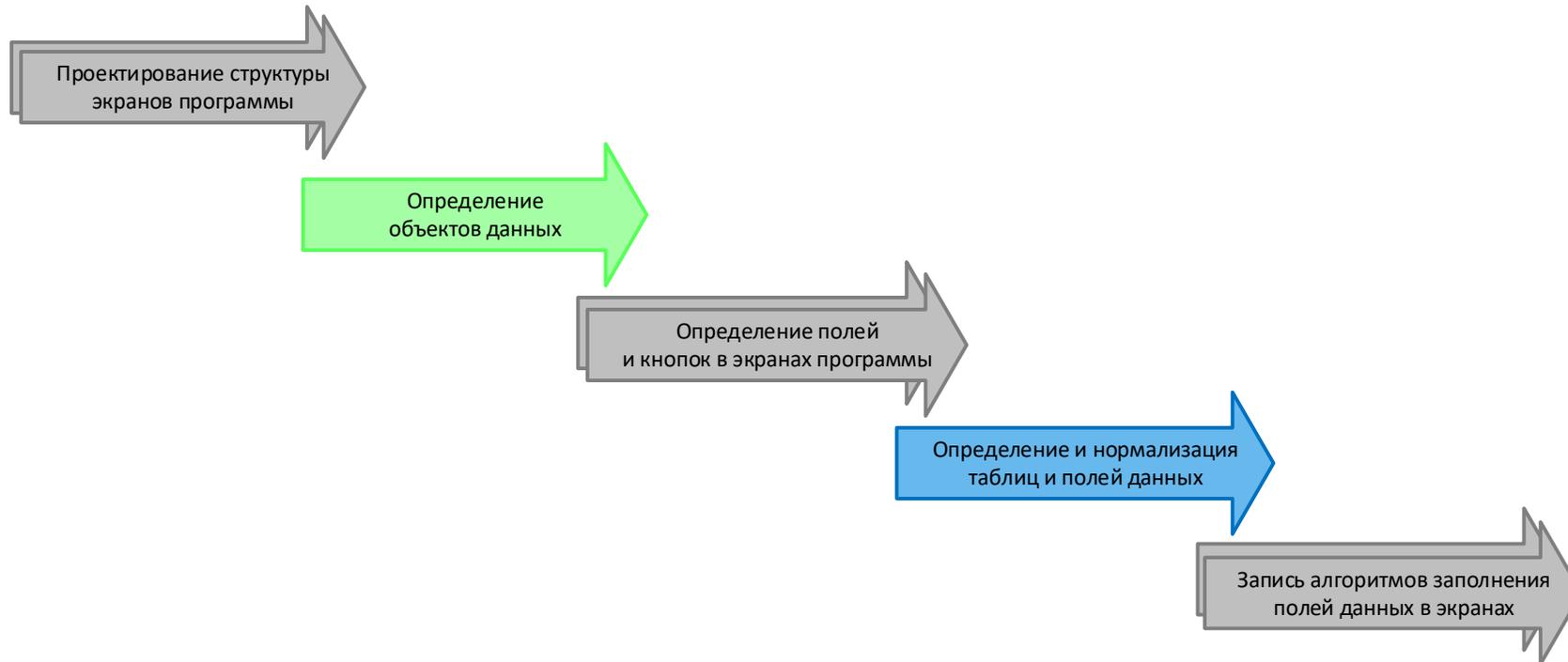
SQL-запрос на русском языке

Выбрать * из ORGANIZATION, где PLANT = «Завод» селекционного экрана и LOCATION = «Склад» селекционного экрана

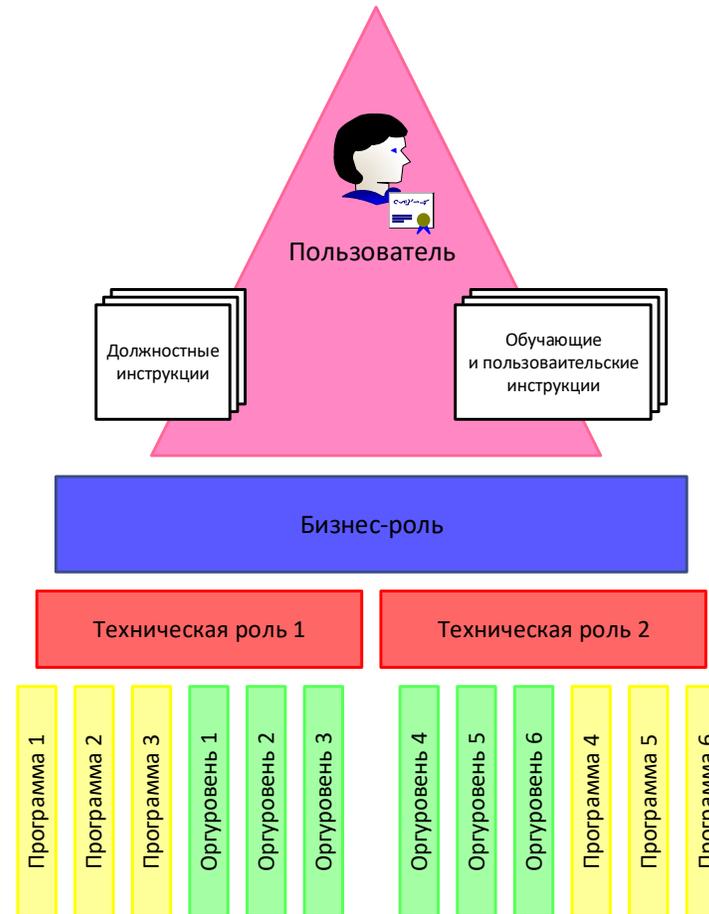
SQL-запрос на английском языке

Select * from ORGANIZATION
where PLANT = «Plant» of selection screen
and LOCATION = «Location» of selection screen

7.22. Шаги проектирования сложных программ



7.23. Роли и полномочия



7.24. Матрица доступа

	Программа 1		Программа 2		Программа 3	
	Запись	Чтение	Запись	Чтение	Запись	Чтение
Роль 1	Да	Да	–	–	–	–
Роль 2	Да	Да	–	Да	Да	Да
Роль 3	–	Да	–	–	–	Да

7.25. Категории тестирования



- 1.1. Безопасности
- 1.2. Взаимодействия
- 1.3. Функциональное
(модульное, интеграционное, системное, приемочное)

1. Функциональное

Категория тестирования программ

2. Нефункциональное



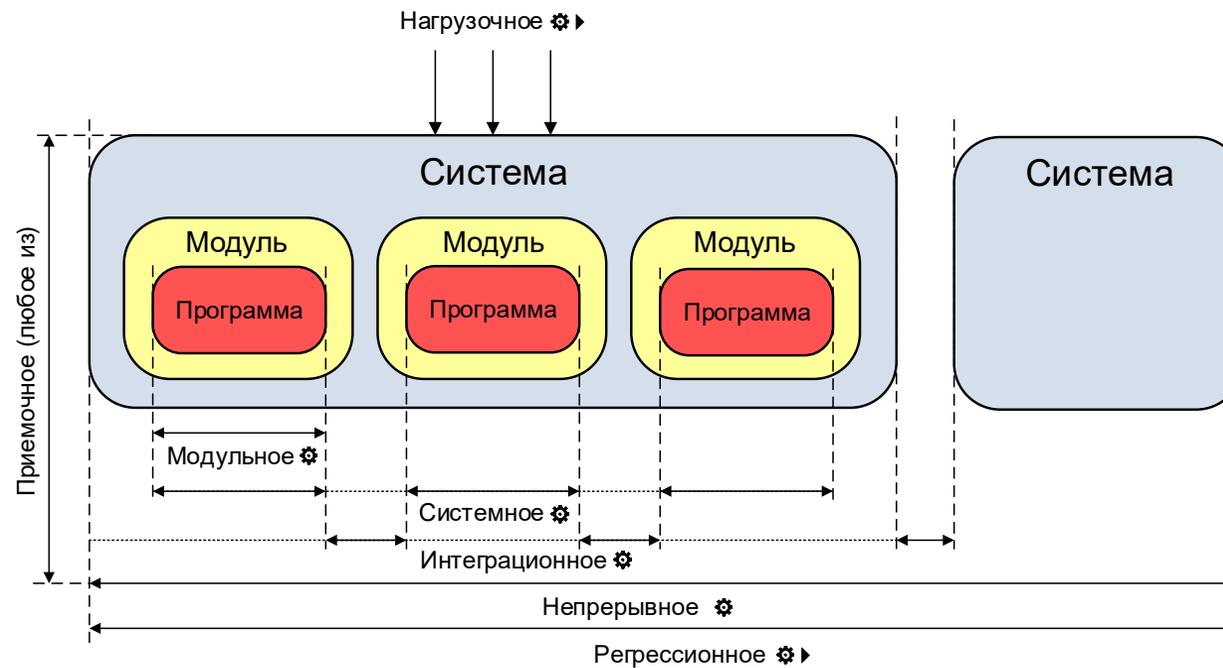
- 2.1. Установки
- 2.2. Удобства использования
- 2.3. Отказа и восстановления
- 2.4. Конфигурационное
- 2.5. Производительности
(нагрузочное, стрессовое, стабильности, объемное)

3. Связанное с изменениями

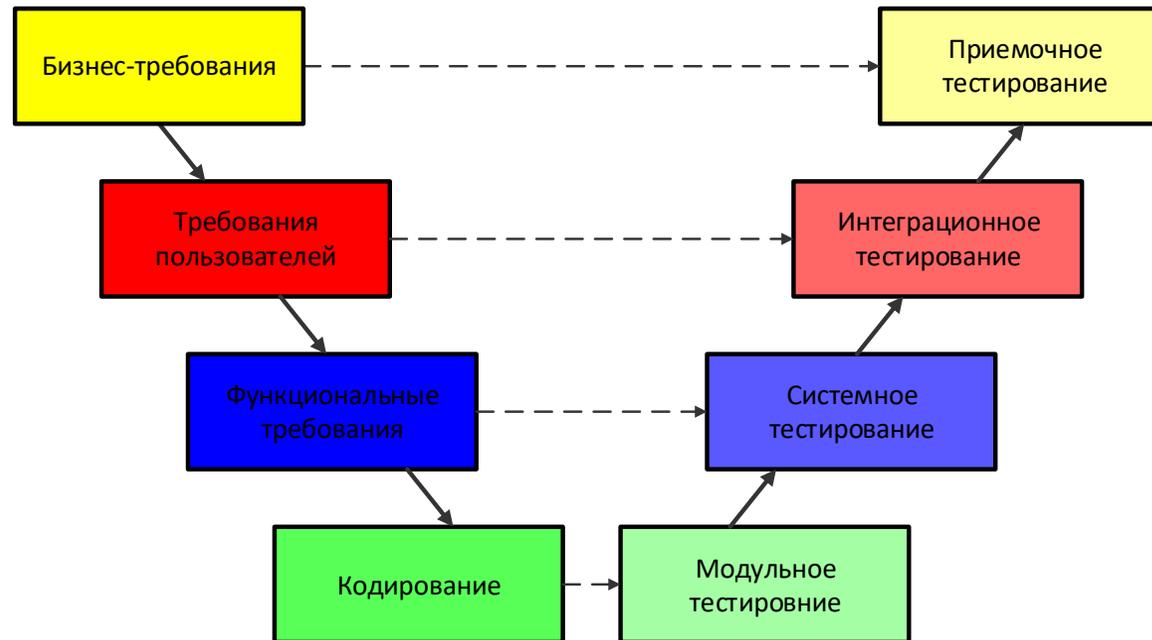


- 3.1. Дымовое
- 3.2. Регрессионное
- 3.3. Сборки
- 3.4. Проверки согласованности

7.26. Виды тестирования



7.27. V-модель разработки через тестирование

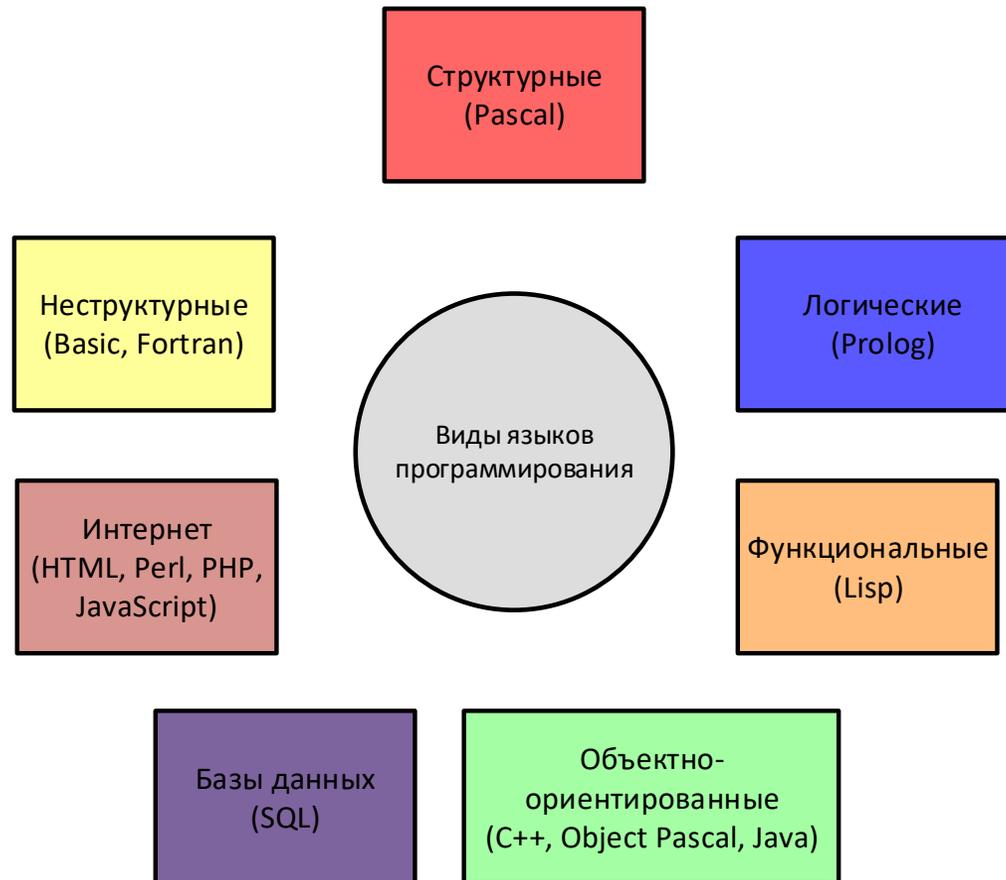


ГЛАВА 8.

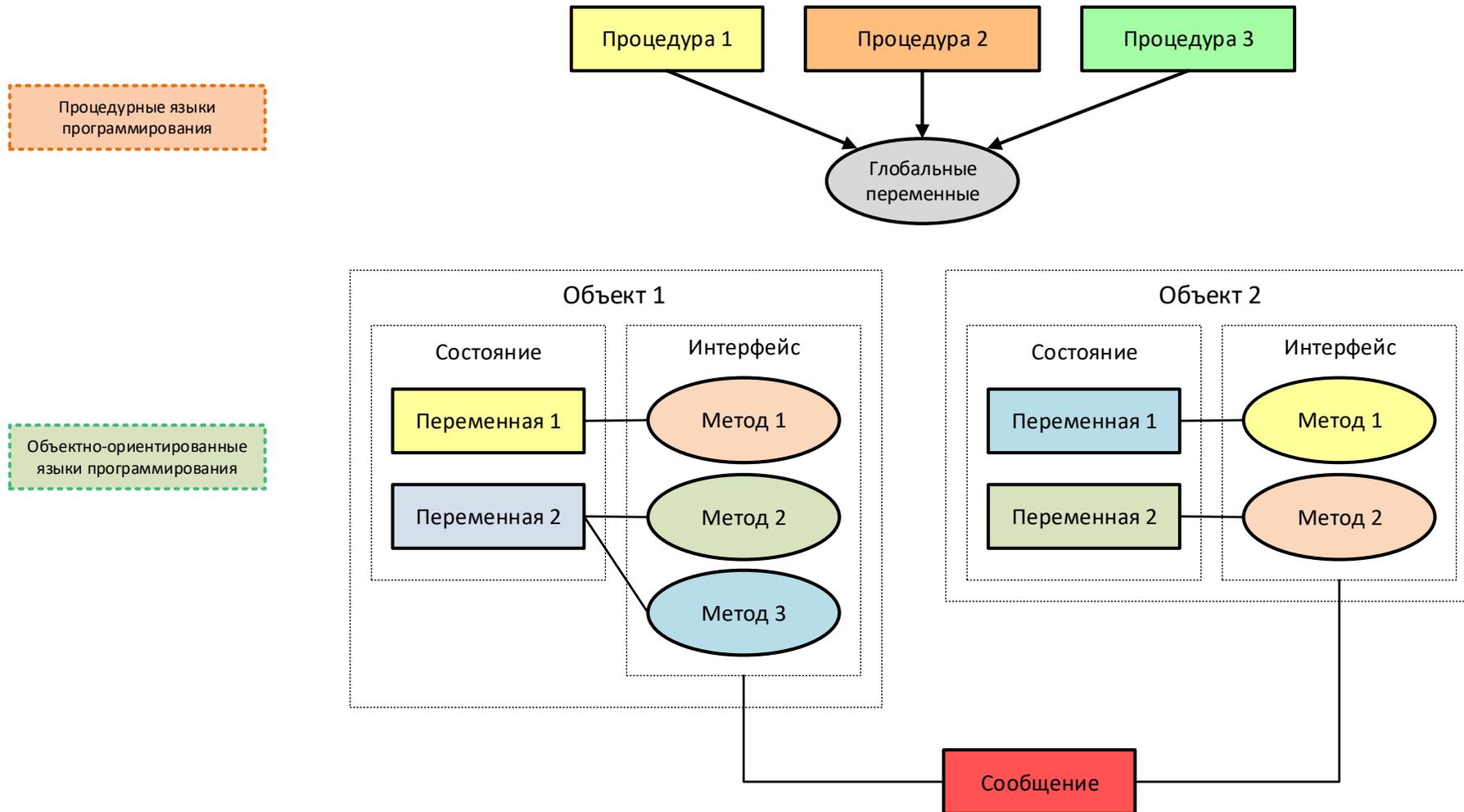
УРОВЕНЬ ПРИЛОЖЕНИЙ И ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

- Виды языков программирования
- Объектно-ориентированные языки программирования
- Объектно-ориентированные и процедурные языки
- Принципы объектно-ориентированного программирования
- Основы объектно-ориентированного программирования C++
- Функции выборки записей и сложные алгоритмы селекции
- Практические примеры

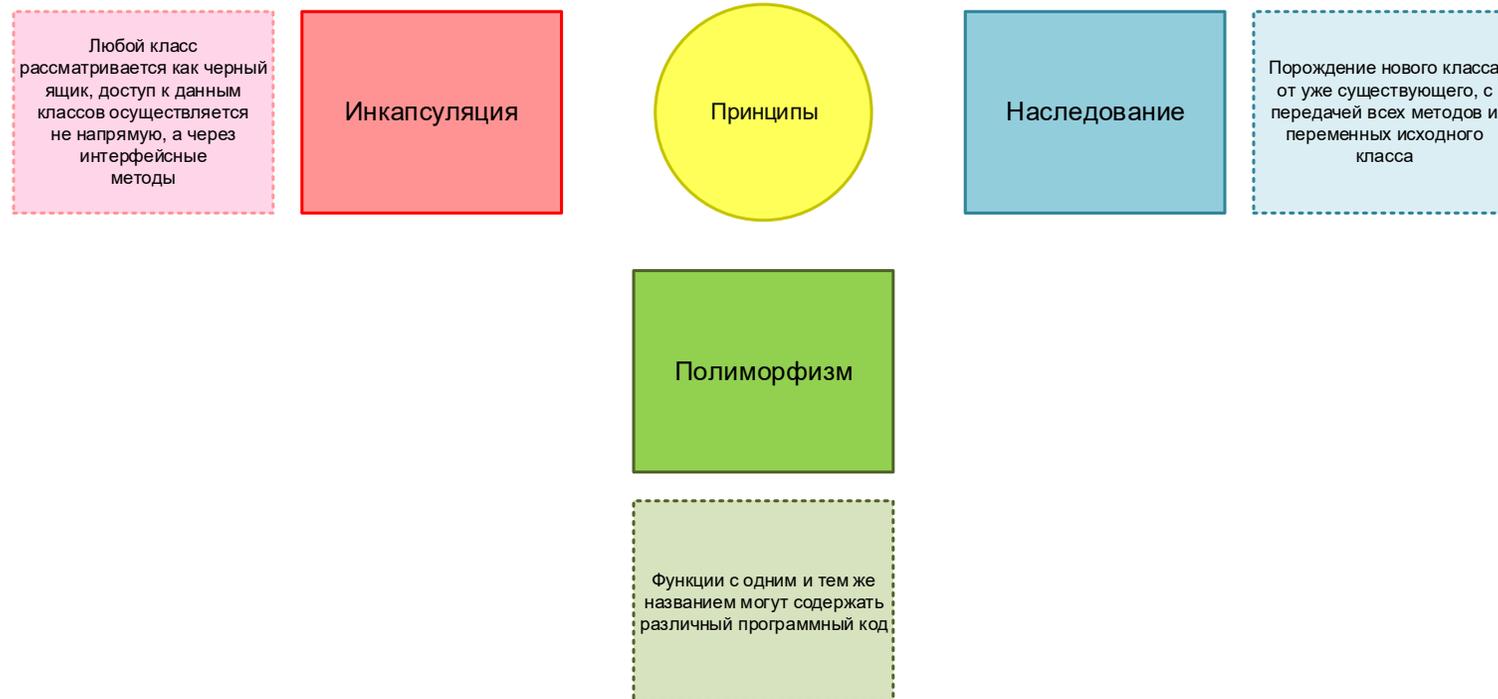
8.2. Виды языков программирования



8.3. Объектные и процедурные языки



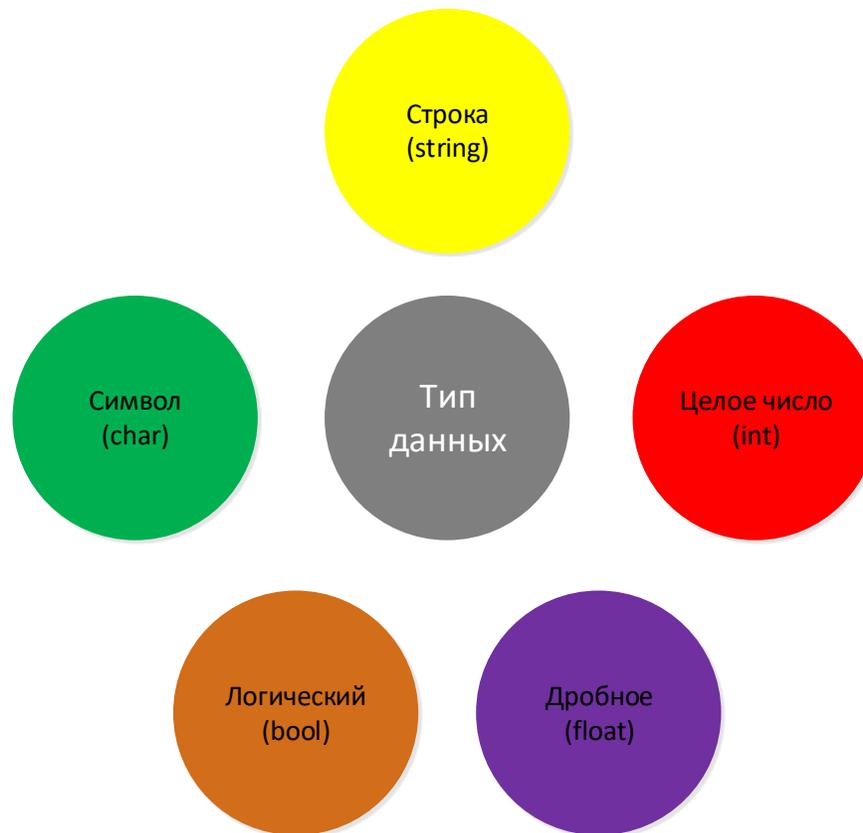
8.4. Принципы объектного программирования



8.5. Основы объектного программирования C++

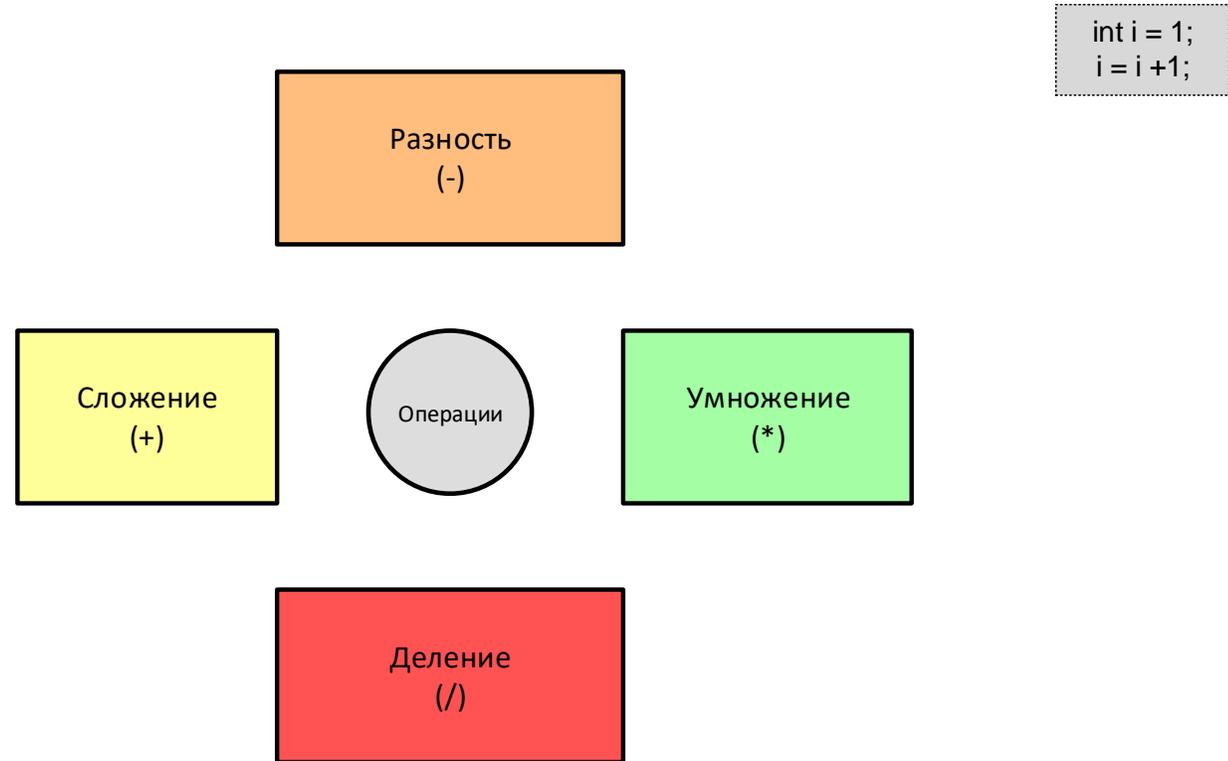


8.6. Типы данных

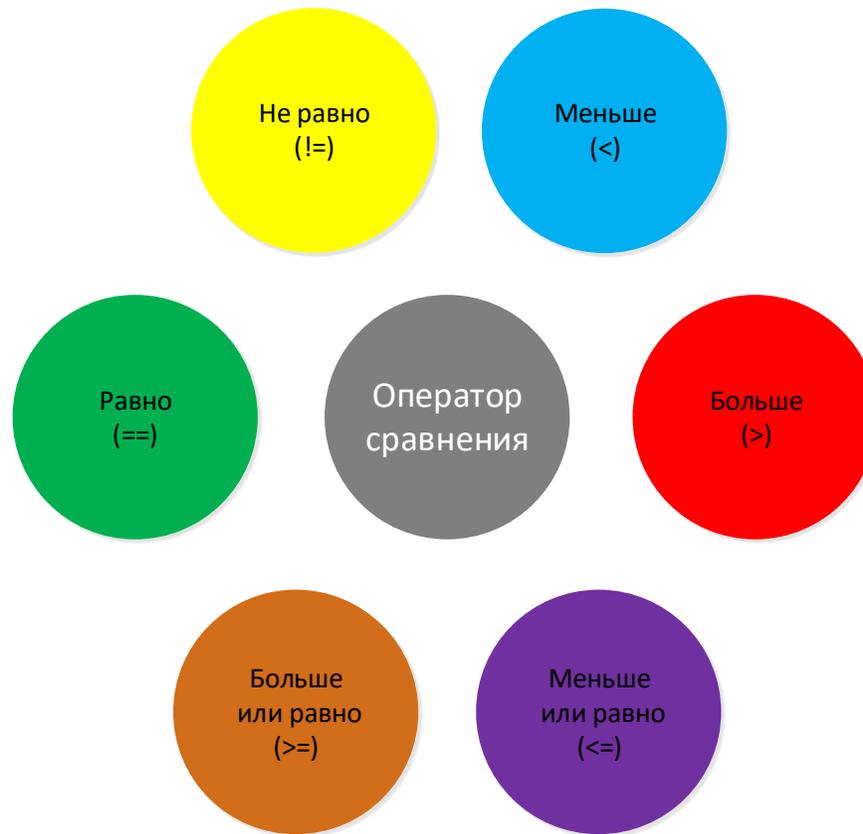


```
int i = 1;
```

8.7. Арифметические операции

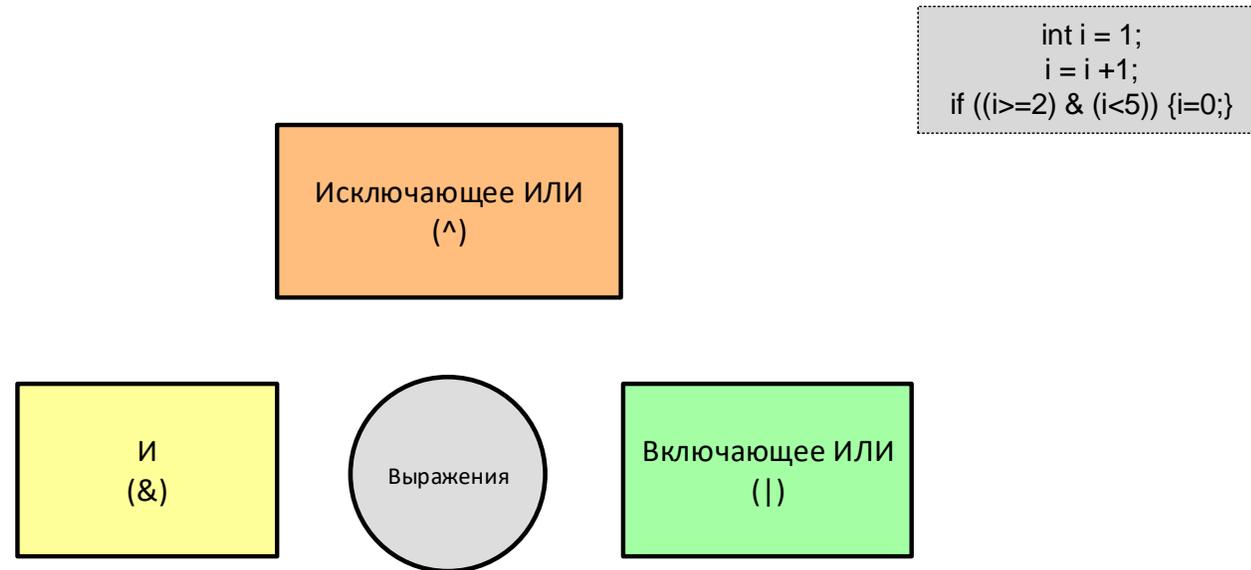


8.8. Операторы сравнения

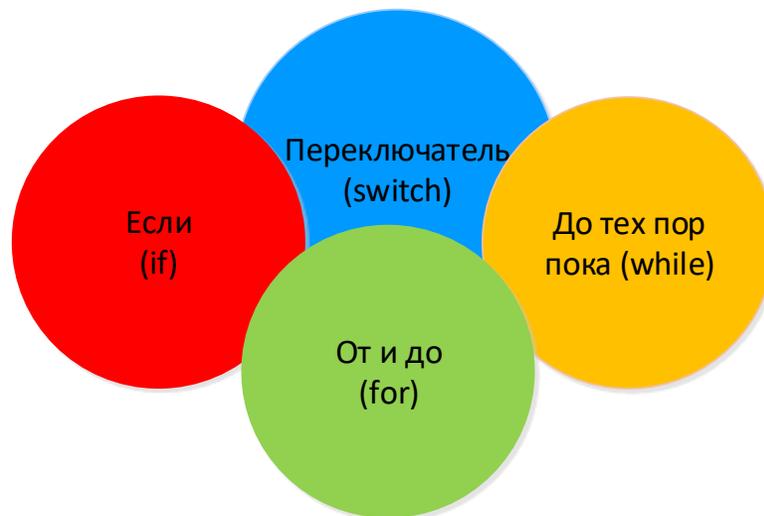


```
int i = 1;  
i = i + 1;  
if (i >= 2) {i = 0;}
```

8.9. Выражения



8.10. Операторы

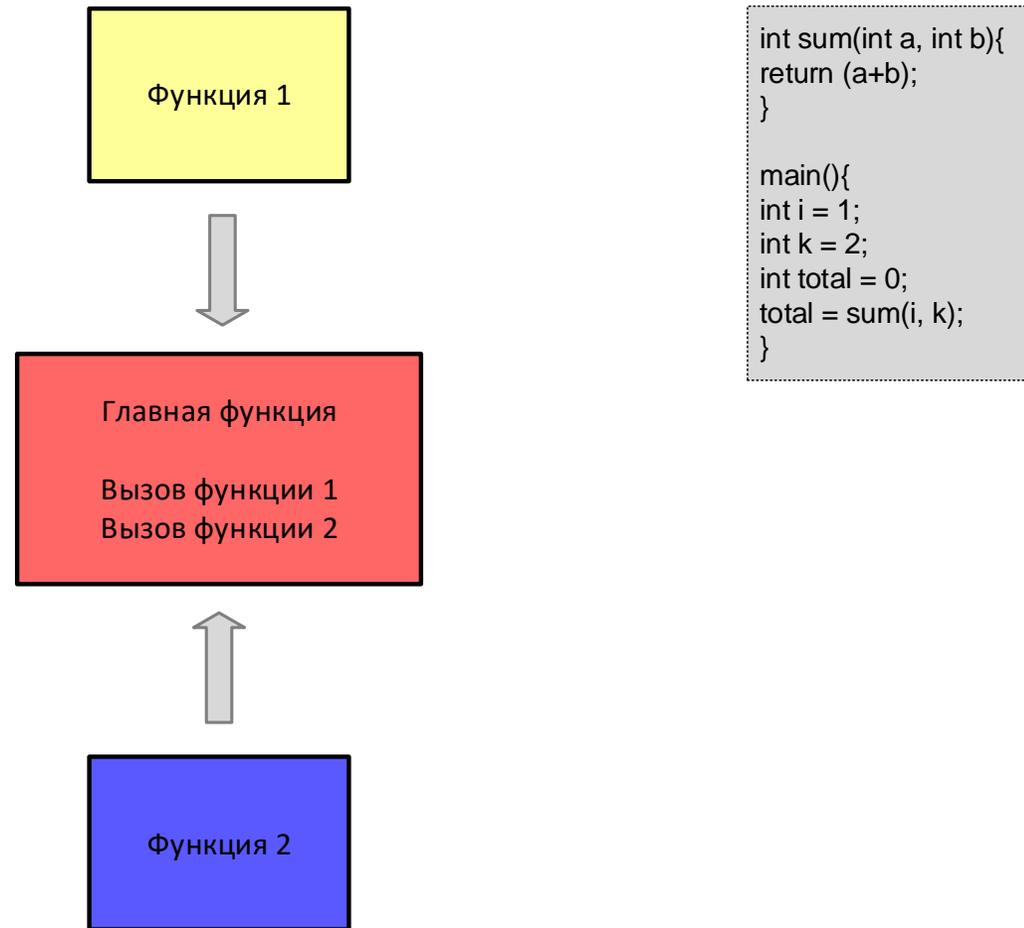


```
char ch = 'a';  
switch (ch){  
  case 'a':  
    ch='x';  
    break;  
  case 'b':  
    ch='y';  
    break;  
  case 'c':  
    ch='z';  
    break;}
```

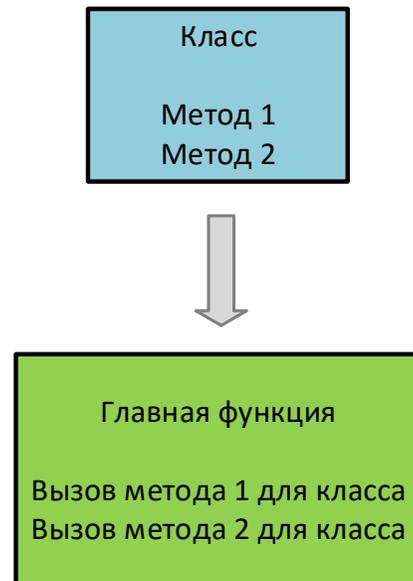
```
int i = 1;  
while (i<5){  
  i=i+1;}
```

```
int i = 1;  
for (int k=0;k<=5;k++){  
  i=i+k;}
```

8.11. ФУНКЦИИ



8.12. Классы

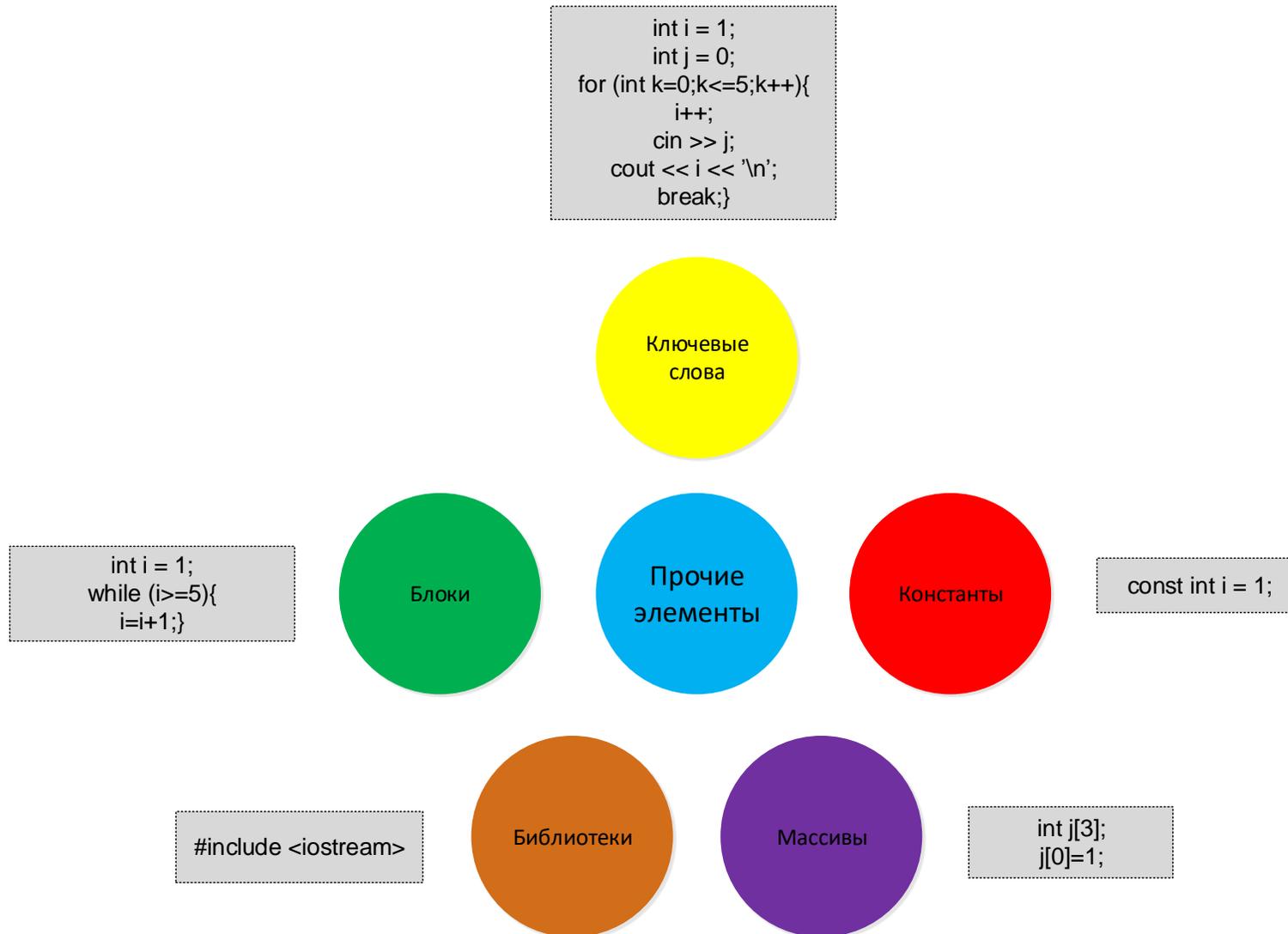


```
class myclass{
private:
int a;
int b;
int c;
public:
void savea(int x){
    a = x;}
void saveb(int y){
    b = y;}
void sumab(){
    c = a+b;}
int returnc(){
    return c;
}
}

main(){
int i = 1;
int k = 2;
int total = 0;
myclass z;

z.savea(i);
z.saveb(k);
z.sumab();
total = z.returnc();
}
```

8.13. Прочие элементы



8.14. Структура программы



```
#include <iostream>

class myclass{
private:
int a;
int b;
int c;
public:
void savea(int x){
    a = x;}
void saveb(int y){
    b = y;}
void sumab(){
    c = a+b;}
int returnc(){
    return c;
}
}

int sum(int a, int b){
return (a+b);
}

main(){
int i = 1;
int k = 2;
int total = 0;
myclass z;

z.savea(i);
z.saveb(k);
z.sumab();
total = z.returnc();
total = sum(i,k);
}
```

8.15. Практический пример 1

Найти наибольшее из 3-х значений,
введенных пользователем

```
Введите первое число:  
1  
Введите второе число:  
14  
Введите третье число:  
7  
Максимальное число:  
14
```

```
#include <iostream>

main(){
int i[3];
int max;

cout << 'Введите первое число:\n';
cin >> i[0];
cout << '\nВведите второе число:\n';
cin >> i[1];
cout << '\nВведите третье число:\n';
cin >> i[2];

max = i[0];
for (int k=1; k<=2; k++){
    if (i[k]>max){
        max = i[k];
    }
}

cout << 'Максимальное число:\n';
cout << max;
}
```

8.16. Практический пример 2

Найти первый номер заданного символа в введенной пользователем строке

Введите строку:
afrbzzrf
Введите символ для поиска в строке:
f
Символ f 2-й слева

Введите строку:
afrbzzrf
Введите символ для поиска в строке:
y
Символ y не найден в строке afrbzzrf

```
#include <iostream>
#include <conio.h>
#include <string.h>
#include <stdio.h>
#include <windows.h>

main(){
char str[256];
int i = -1;
char ch;

cout << 'Введите строку:\n';
cin >> str;
cout << '\nВведите символ для поиска в строке:\n';
cin >> ch;
for (int k=0; k<strlen(str); k++){
    if (str[k]==ch){
        i = k+1;
        break;
    }
}

If (i>0){
    cout << '\nСимвол' << ch << i << '-й слева';
}
else{
    cout << '\nСимвол' << ch << 'не найден в строке' << str;
}
}
```

8.17. Практический пример 3

Вывести название дня недели
по номеру дня, введенному
пользователем

Введите номер дня недели:
1
Понедельник

Введите номер дня недели:
z
День не определен

```
#include <iostream>

main(){
char ch;
char day[20] = 'День не определен';

cout << 'Введите номер дня недели:\n';
cin >> ch;
switch (ch){
    case '1':
        day='Понедельник';
        break;
    case '2':
        day='Вторник';
        break;
    case '3':
        day='Среда';
        break;
    case '4':
        day='Четверг';
        break;
    case '5':
        day='Пятница';
        break;
    case '6':
        day='Суббота';
        break;
    case '7':
        day='Воскресенье';
        break;
    }
cout << '\n' <<day;
}
```