

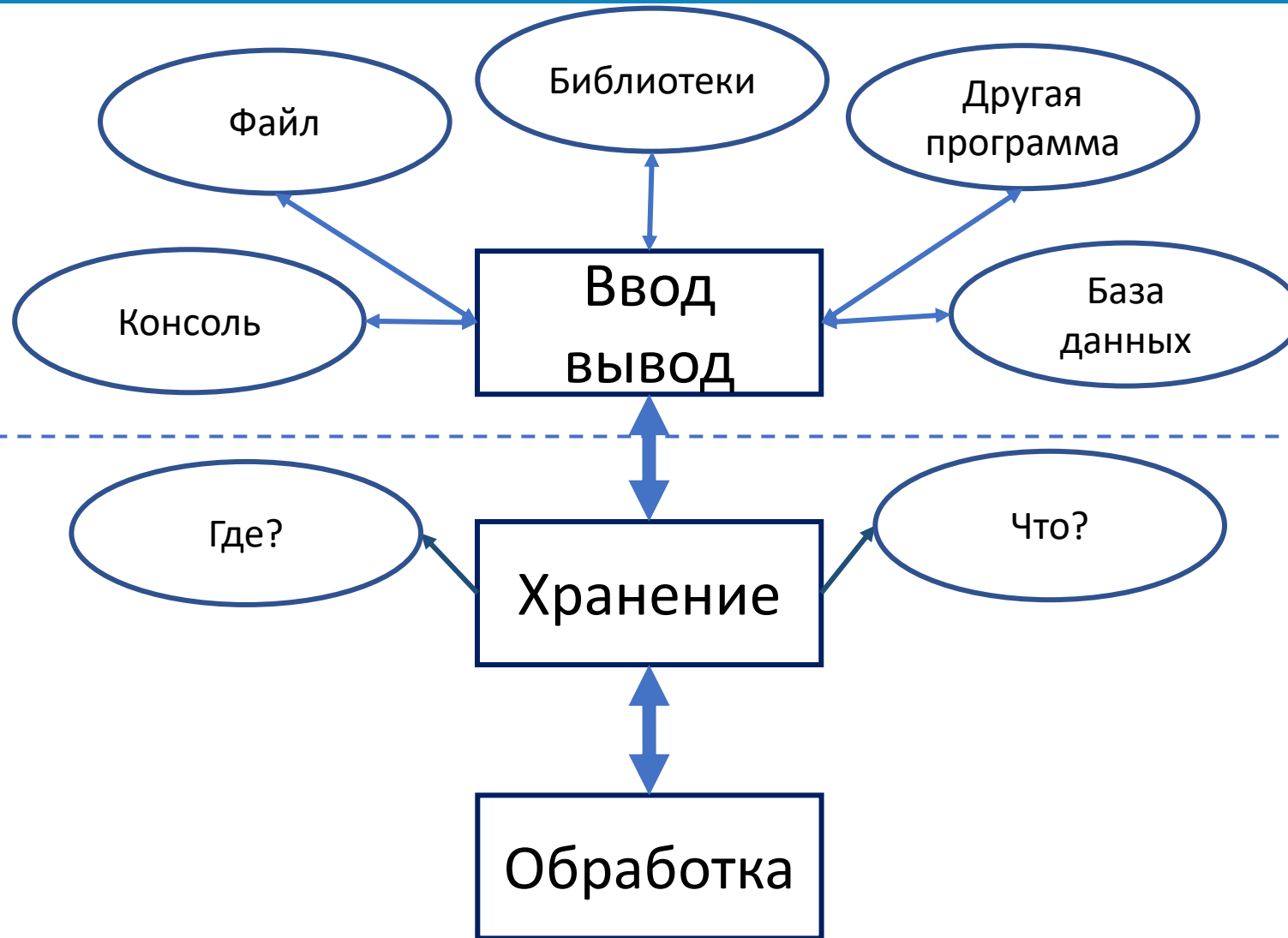


Основы программирования на C++

Занятие 8. Введение в язык программирования C++

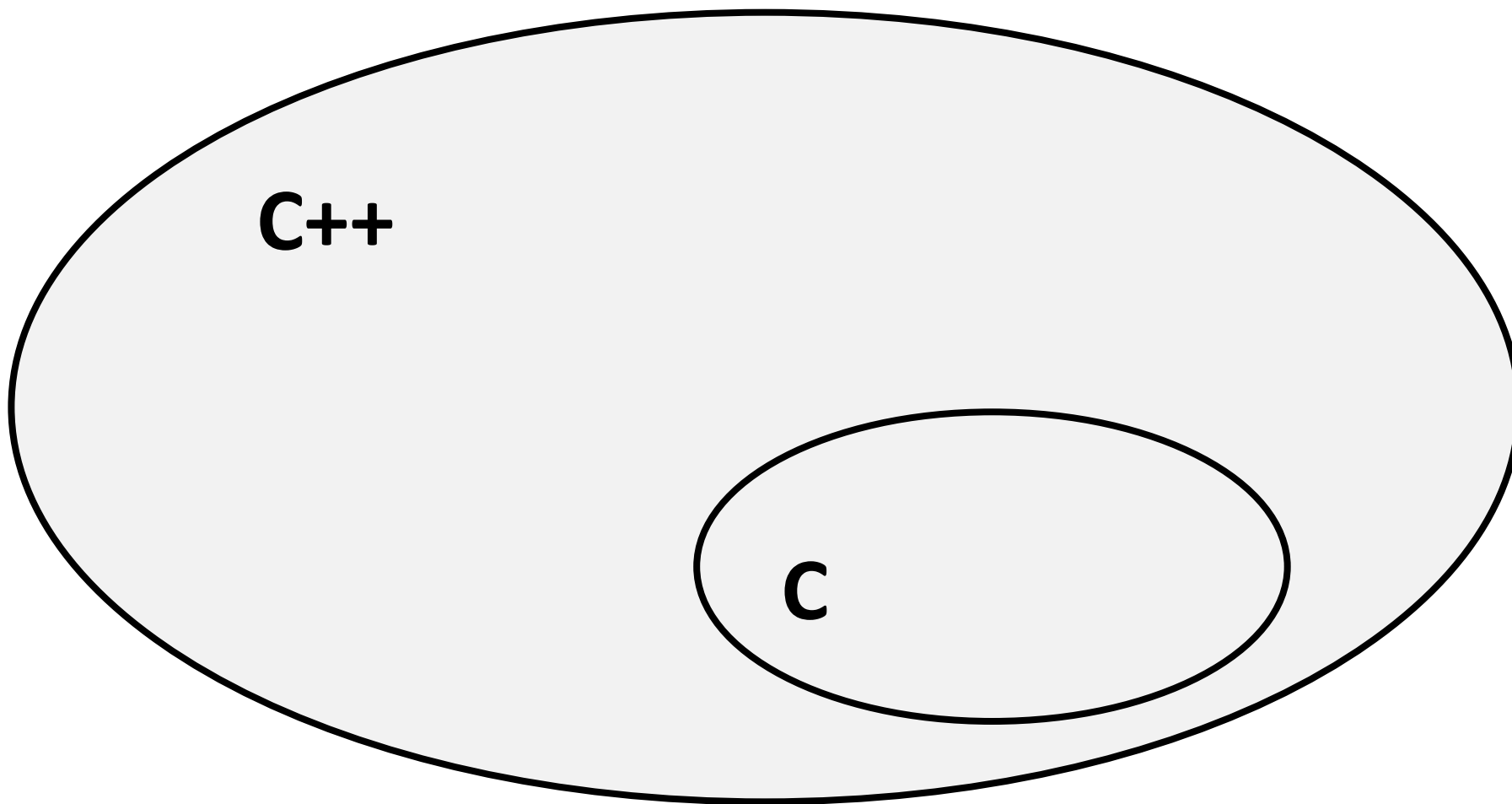


Дерево языка





ОСНОВЫ C++





Пример программы

C

```
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
}
```

C++

```
#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}
```



Пространство имен

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    cout << "Hello World!\n";
```

```
}
```

Пространство имен — это декларативная область, в рамках которой определяются различные идентификаторы (имена типов, функций, переменных, и т. д.).



Пространства имен

```
#include <iostream>

namespace Summ
{
    int Func(int a, int b) {
        return a + b;
    }
}

namespace Mult
{
    int Func(int a, int b) {
        return a * b;
    }
}

int main()
{
    std::cout << "Mult = " << Mult::Func(5, 5) << std::endl;
    std::cout << "Summ = " << Summ::Func(5, 5) << std::endl;
}
```

```
Mult = 25
Summ = 10
```



Потоки ввода/вывода

C

```
#include <stdio.h>

int main()
{
    int x;
    scanf_s("%d", &x);
    printf("X = %d\n", x);
    return 0;
}
```

C++

```
#include <iostream>

int main()
{
    int x;
    std::cin >> x;
    std::cout << "X = " << x << std::endl;
}
```



Выделение динамической памяти

C

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int N;
    scanf_s("%d", &N);
    int* p = (int*)malloc(N * sizeof(int));
    for (size_t i = 0; i < N; ++i)
    {
        p[i] = i;
    }
    for (size_t i = 0; i < N; ++i)
    {
        printf("%d ", p[i]);
    }
    printf("\n");
    free(p);
    return 0;
}
```

C++

```
#include <iostream>

int main()
{
    int N;
    std::cin >> N;
    int* p = new int[N];
    for (size_t i = 0; i < N; ++i)
    {
        p[i] = i;
    }
    for (size_t i = 0; i < N; ++i)
    {
        std::cout << p[i] << " ";
    }
    std::cout << std::endl;
    delete []p;
    return 0;
}
```




Ссылки

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

void func_1(int& x){
    x++;
}

void func_2(int x){
    x++;
}

void main()
{
    int a = 10;
    cout << a << endl;
    func_1(a);
    cout << a << endl;
    func_2(a);
    cout << a << endl;
}
```

Ссылка в C++ -- это альтернативное имя объекта.

```
10
11
11
```



Задача

Необходимо добавить элемент в конец массива



Возможные решения - 1

1. Создать новый массив размера $N+1$, скопировать в него старый массив и добавить на $N+1$ место данный элемент
2. Воспользоваться функцией `realloc`.



Возможные решения - 2

1. Создать новый массив размера $N+1$, скопировать в него старый массив и добавить на $N+1$ место данный элемент
2. Воспользоваться функцией `realloc`.



Вектора C++

```
#include <iostream>
#include <vector>

int main()
{
    std::vector<int> v;
    int length = 5;
    std::cin >> length;
    for (size_t i = 0; i < length; i++)
    {
        v.push_back(rand() % 100);
    }
    for (size_t i = 0; i < v.size(); i++)
    {
        std::cout << v[i] << " ";
    }
    std::cout << std::endl;
    return 0;
}
```

Вектор в C++ — это замена стандартному динамическому массиву, память для которого выделяется вручную, с помощью оператора new.

Полный код см. в заметках к слайду

Спасибо за внимание!