1. **Функции. Объявление, определение, вызов. Понятия прототипа и сигнатуры. Способы передачи аргументов в функцию. Типы возвращаемого значения и оператор return. Функции без возвращаемого значения. Рекурсивные функции.**

**Функция** - некоторый паттерн команд.

* *Объявление* — указывание типа возвращаемого значения, имени функции и аргументов
* *Определение* — прописывание тела функции, что делать при вызове.
* *Вызов* — обращение к выполнению функции.

**Прототип** - объявление функции без её определения, просто шапка, чтобы компилятор понимал, что такая функция существует и она где-то определена.

**Сигнатура** - логическое понятие, по сути это то, какие типы аргументов существуют могут передаваться в функцию и их порядке — это то, что различает функции при перегрузке.

* Способы передачи аргументов в функцию

**По значению**: void func(int a); - создается локальная копия передаваемой переменной.

**По ссылке**: void func(int& a); - используется та же переменная, что и передается, только с именем а(как второе имя для той же ячейки)

**По указателю**: void func(int \* a); и при вызове func(&a); - создается указатель на int, в которой хранится адрес переменной а.

* **Типы возвращаемого значения**

type func(...){

return type\_var; - возвращение того же типа, что и указан в прототипе.

}

**Функция без возвращаемого значения** - в типе пишется слово void и return не нужен.

**Рекурсии** - вызов в функции самой себя. Главное, чтобы она могла остановиться после выполнения определенного условия, чтобы избежать переполнения памяти.