 **Язык Java. Особенности языка.**

* Java — это объектно-ориентированный, строго типизированный, кросс-платформенный язык программирования. Главные особенности:
  + Платформонезависимость: программы на Java компилируются в байт-код, который исполняется JVM (Java Virtual Machine).
  + Автоматическое управление памятью (Garbage Collection).
  + Безопасность, многопоточность и поддержка распределенных вычислений.

 **Средства разработки. JDK и JRE. Компиляция и выполнение программы. JAR-архивы.**

* **JDK (Java Development Kit)** — комплект для разработки, включающий компилятор javac, библиотеки, утилиты и интерпретатор.
* **JRE (Java Runtime Environment)** — среда выполнения, необходимая для запуска Java-программ. Включает JVM и стандартные библиотеки.
* **Компиляция программы:** происходит с помощью компилятора javac, который преобразует исходный код в байт-код.
* **JAR-архивы** — это Java ARchive файлы, которые содержат скомпилированные классы и метаданные для удобства распространения приложений.

 **Примитивные типы данных в Java. Приведение типов.**

* В Java есть 8 примитивных типов данных:
  + Целочисленные: byte, short, int, long
  + Числа с плавающей запятой: float, double
  + Логический тип: boolean
  + Символьный тип: char
* **Приведение типов** может быть явным (кастинг, например (int)) или неявным (автоматическое преобразование в сторону увеличения типа).

 **Работа с переменными. Декларация. Инициализация. Присваивание.**

* **Декларация** переменной: объявление переменной с указанием типа, например: int x;.
* **Инициализация**: присваивание начального значения при создании переменной, например: int x = 10;.
* **Присваивание**: изменение значения переменной после декларации, например: x = 20;.

 **Одномерные и двумерные массивы. Декларация и создание массивов. Доступ к элементам массива.**

* **Одномерный массив**: int[] arr = new int[5];
* **Двумерный массив**: int[][] matrix = new int[3][4];
* **Доступ к элементам** осуществляется с помощью индексов, начиная с нуля: arr[0], matrix[2][3].

 **Инструкции ветвления (if-else, switch) и циклов (do, while, for).**

* **if-else**: условное выполнение кода.
* **switch**: выбор одной из нескольких ветвей в зависимости от значения выражения.
* **Циклы**:
  + for: цикл с известным числом итераций.
  + while: цикл с проверкой условия в начале.
  + do-while: цикл с проверкой условия в конце.

 **Операторы и выражения в Java. Особенности вычисления, приоритеты операций.**

* Операторы: арифметические (+, -, \*, /, %), логические (&&, ||, !), битовые, операторы сравнения.
* **Приоритеты операций** определяют последовательность выполнения (например, сначала умножение, потом сложение).
* В Java также поддерживаются сокращенные операторы, например +=.

 **Математические функции в составе стандартной библиотеки Java. Класс java.lang.Math.**

* Класс Math содержит статические методы для выполнения математических операций, таких как Math.sqrt(), Math.pow(), Math.sin(), Math.random(), Math.abs() и другие.

 **Подпрограммы, методы, параметры и возвращаемые значения.**

* **Методы** — это подпрограммы, которые выполняют задачи и могут возвращать значения.
* **Параметры** методов могут передаваться по значению. Метод может возвращать результат с помощью return.
* Пример: public int sum(int a, int b) { return a + b; }

 **Форматированный вывод числовых данных.**

* Используется класс System.out.printf() для форматирования вывода.
* Пример: System.out.printf("Число с двумя знаками после запятой: %.2f", 3.14159);