- Создать интерфейс `Processor` с методом `process(T obj)` для работы с объектом.

- В методе `main` создать **лямбду**, которая реализует этот интерфейс для обработки/вывода объектов.

Практическое занятие №5 (делегаты + структуры)

Тема: Работа с типом данных *структура* и делегаты

Общий план выполнения:

1. Создать класс (структуру) с полями по условию.

2. Создать функциональный интерфейс `Processor<T>`:

```
@FunctionalInterface
interface Processor<T> {
    void process(T obj);
}
```

3. Создать массив объектов.

4. Создать лямбду (делегат) типа `Processor`, которая будет обрабатывать объект (например, выводить, фильтровать, считать).

5. Использовать делегат для всех объектов по условию (например, выше среднего, отфильтрованные).

1️⃣ Вариант

Student: `name`, `age`, `averageGrade`.

- Найти студентов с баллом выше среднего.

- Использовать делегат `Processor<Student>` для **вывода** каждого подходящего студента.

2️⃣ Вариант

Book: `title`, `author`, `price`.

- Найти книги дороже средней цены, отсортировать по цене.
- Использовать делегат для **вывода информации о книге**.

3️⃣ Вариант

Car: `brand`, `year`, `mileage`.

- Найти машины новее 2015 года с пробегом < 50 000, отсортировать по пробегу.
- Делегат для вывода/форматирования строки с информацией о машине.

4️⃣ Вариант

Employee: `name`, `position`, `salary`.

- Найти сотрудников с зарплатой выше средней, отсортировать по убыванию.
- Делегат для **вывода сотрудника**.

5️⃣ Вариант

Product: `name`, `price`, `quantity`.

- Найти товары с общей стоимостью выше средней.
- Делегат для **вывода каждого товара**.

6️⃣ Вариант

Movie: `title`, `rating`, `year`.

- Найти фильмы с рейтингом > 8.0, отсортировать по году.
- Делегат для вывода фильма.

7️⃣ Вариант

City: `name`, `population`, `area`.

- Плотность населения выше средней, отсортировать по плотности.
- Делегат для вывода информации о городе.

8️⃣ Вариант

Course: `title`, `duration`, `price`.

- Найти курсы с ценой за час ниже среднего, отсортировать по цене за час.
- Делегат для вывода курса.

9️⃣ Вариант

Laptop: `brand`, `ram`, `price`.

- Найти ноутбуки с RAM ≥ 16GB, выбрать два самых дешёвых.
- Делегат для вывода информации о ноутбуке.

🔟 Вариант

Song: `title`, `artist`, `duration`.

- Песни с длительностью больше средней, отсортировать по возрастанию.
- Делегат для вывода песни.

1️⃣1️⃣ Вариант

SportTeam: `teamName`, `gamesWon`, `gamesLost`.

- Вычислить процент побед > 70%, отсортировать по проценту побед.
- Делегат для вывода команды.

1️⃣2️⃣ Вариант

Plane: `model`, `seats`, `range`.

- Самолёты с дальностью ниже средней, отсортировать по возрастанию.

- Делегат для вывода информации о самолёте.

1️⃣3️⃣ Вариант

Recipe: `dishName`, `calories`, `cookingTime`.

- Блюда с калорийностью < 500 и временем < 30 минут, отсортировать по времени.
- Делегат для вывода рецепта.

1️⃣4️⃣ Вариант

Phone: `model`, `storage`, `price`.

- Отсортировать по цене за гигабайт, вывести три лучших.
- Делегат для вывода телефона.

1️⃣5️⃣ Вариант

Order: `orderId`, `itemCount`, `totalCost`.

- Заказы дороже средней стоимости, отсортировать по убыванию.
- Делегат для вывода заказа.

1️⃣6️⃣ Вариант

Pet: `name`, `species`, `age`.

- Питомцы старше 5 лет, отсортировать по возрасту.
- Делегат для вывода информации о питомце.

1️⃣7️⃣ Вариант

Building: `address`, `floors`, `yearBuilt`.

- Построенные после 2010 года, вывести три последних.
- Делегат для вывода информации о здании.

1️⃣8️⃣ Вариант

University: `name`, `studentsCount`, `rating`.

- Рейтинг выше среднего, отсортировать по количеству студентов.
- Делегат для вывода университета.

1️⃣9️⃣ Вариант

Game: `title`, `genre`, `playersOnline`.

- Онлайн > 10 000, отсортировать по убыванию.
- Делегат для вывода информации об игре.

2️⃣0️⃣ Вариант

Delivery: `destination`, `distance`, `cost`.

- Цена за км ниже средней, отсортировать по возрастанию.
- Делегат для вывода доставки.