**Суть за 30 секунд**

* **Weapon** — общий шаблон оружия (как «родитель»): у каждого есть имя, урон и вес. Умеет **атаковать** и считает, сколько всего оружий создано.
* **MeleeWeapon** и **RangedWeapon** — это **категории** (второй уровень): ближний бой и дальний бой.
* **Sword**, **Bow**, **MagicWand** — **конкретные** виды оружия, каждое атакует по-своему.
* **Main** — консольное меню: посмотреть список, добавить оружие, атаковать по цели, посчитать DPS, показать счётчик.

**Карта «что в ТЗ» → «где в коде»**

* **Абстрактный базовый класс (Оружие)** → Weapon (abstract, метод attack() абстрактный).
* **2 уровня наследования** →  
  Weapon → MeleeWeapon → Sword  
  Weapon → RangedWeapon → Bow / MagicWand
* **Инкапсуляция (скрытые поля)** → все поля private, доступ через геттеры/сеттеры (getDamage(), setDamage() и т. п.).
* **Перегрузка** (одна операция, разные параметры) → Weapon.attack(), attack(String), attack(String,int).
* **Переопределение** (свой вариант в наследнике) → Sword/Bow/MagicWand переопределяют attack(); Bow и MagicWand переопределяют dps(...).
* **Конструкторы (в т.ч. по умолчанию)** → есть без параметров и с параметрами в каждом классе; в наследниках — вызов super(...) первым.
* **Геттеры/сеттеры** → во всех классах.
* **Ввод/вывод** → в Main: считываем из консоли через Scanner, печатаем через System.out.println.
* **Статический счётчик объектов** → поле в Weapon, растёт в конструкторе; показываем через Weapon.getCreatedCount().

**Как это «чувствуется» при запуске**

1. Запускаешь Main → видишь меню.
2. «1» — показывает все созданные мечи/луки/палочки.
3. «2/3/4» — добавляет новый меч/лук/палочку (ты вводишь параметры).
4. «5» — **полиморфизм**: у нас один список List<Weapon>, но attack() у каждого вида работает по-разному (меч рубит, лук стреляет, палочка тратит ману).
5. «6» — показывает **DPS**: общий метод, но лук/палочка считают по-своему.
6. «7» — печатает **сколько всего** оружий создано (работа статического счётчика).

**Простой словарик (чтобы объяснить преподавателю)**

* **Наследование**: «дети» (Sword, Bow, MagicWand) берут общее из «родителя» (Weapon) и добавляют своё.
* **Инкапсуляция**: поля спрятаны (private), доступ только через методы (без прямых «пальцев в кишки»).
* **Перегрузка**: одна команда attack, но с разными аргументами (без цели / с целью / много раз).
* **Переопределение**: общий attack() есть у всех, но реализация у каждого своя.
* **Полиморфизм**: храним всё как Weapon, а ведут себя по-разному.

**Если спросят «покажи на примере»**

* Введи «5», цель: Скелет.  
  Увидишь: меч — «⚔️ удар», лук — «🏹 выстрел (минус стрелы)», палочка — «✨ заклинание (минус мана)».  
  Это и есть полиморфизм вживую.

**Что сказать «про будущее» (если попросят развить)**

* Легко добавить новый вид оружия: создать класс Dagger extends MeleeWeapon и реализовать свой attack(). Всё меню уже будет работать, потому что хранение идёт через List<Weapon>.

Если нужно — отмечу комментариями в коде строки, на которые ты будешь ссылаться при защите.

**Программа «Оружие»**

**Что это такое:** маленький симулятор арсенала. Вы собираете коллекцию оружия и «атакуете» цель, чтобы посмотреть, как оно себя ведёт.

**Что вы делаете в меню:**

* **1) Показать арсенал** — видите список уже созданных предметов (мечи, луки, палочки) с краткими характеристиками.
* **2) Добавить Меч / 3) Добавить Лук / 4) Добавить Волшебную палочку** — программа задаёт простые вопросы (урон, вес и т.п.) и добавляет новый предмет в список.
* **5) Атаковать всеми по цели** — вводите, кто «противник» (например, «скелет»). Каждый предмет выводит свой текст атаки:
  + меч — размах и урон,
  + лук — выстрел и сколько стрел осталось,
  + палочка — заклинание и сколько маны осталось.  
    В конце программа повторяет атаку последним предметом несколько раз.
* **6) Показать DPS** — грубая оценка «урона в секунду» для каждого предмета при одинаковом темпе действий.
* **7) Показать счётчик** — сколько всего предметов вы уже создали.
* **0) Выход** — закрыть программу.

**Зачем это вам как пользователю:** быстро «пощупать» разные типы оружия, добавлять свои варианты и наблюдать результат действий (сообщения в консоли: выстрелы, трата маны, урон, остаток стрел).