**Лабораторная работа 1.**

**Задание 1**

Необходимо создать три класса, которые будут наследоваться от базового класса GameObject, и описать их соответствующие содержание:

* Персонаж – имя, очки здоровья, очки опыта, текущий уровень, силу атаки и уровень защиты.
* Противник – название, очки здоровья, силу атаки и уровень защиты.
* Оружие – название, силу атаки, стоимость, требуемый уровень для персонажа.

В любом классе создать один метод, который возвращает любое соответствующее свойство.

**Результат**:

Был создан класс GameObject, от которого наследуются классы Hero (персонаж), Enemy (противник), Weapon (оружие). В классе Hero был создан метод public int getHP(), который возвращает значение здоровья персонажа.

class GameObject {  
 protected GameObject() {}  
 protected void show() {}  
}  
  
class Hero extends GameObject {  
 private String name;  
 private int hp, exp, level, atk, deff;  
  
 public Hero(String name, int hp, int exp, int level, int atk, int deff) {  
 this.name = name;  
 this.hp = hp;  
 this.exp = exp;  
 this.level = level;  
 this.atk = atk;  
 this.deff = deff;  
 }  
  
 public Hero() {  
 name = "Hero";  
 hp = exp = level = atk = deff = 0;  
 }  
  
 public int getHp() {  
 return hp;  
 }  
  
}  
  
class Enemy extends GameObject {  
 private String name;  
 private int hp, atk, deff;  
  
 public Enemy(String name, int hp, int atk, int deff) {  
 this.name = name;  
 this.hp = hp;  
 this.atk = atk;  
 this.deff = deff;  
 }  
  
 public Enemy() {  
 name = "Enemy";  
 hp = atk = deff = 0;  
 }  
  
}  
  
class Weapon extends GameObject {  
 private String name;  
 private int attack, coast, need\_level;  
  
 public Weapon(String name, int attack, int coast, int need\_level) {  
 this.name = name;  
 this.attack = attack;  
 this.coast = coast;  
 this.need\_level = need\_level;  
 }  
  
 public Weapon() {  
 name = "Weapon";  
 attack = coast = need\_level = 0;  
 }  
  
}

**Задание 2**

В классе “персонаж” создать метод, который по полученному значению требуемого уровня оружия проверяет может ли игрок его экипировать (выводит сообщение на экран).

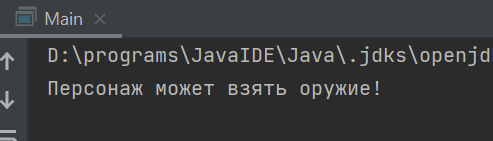
**Результат**

В классе Hero был создан метод, который при передаче ему параметра выполняет условия задания:

public void takeAWeapon(int level\_weapon) {  
  
 if (level >= level\_weapon) {  
 System.*out*.println("Персонаж может взять оружие!\n");  
 } else {  
 System.*out*.println("Персонаж не может поднять оружие!\n");  
 }  
}

По умолчанию был создан персонаж со здоровьем (hp) равному 77, в метод был передан параметр оружия равный так же 77.

Скриншот выполнения работы:



**Задание 3**

Создать два объектных массива: один с объектами типа “Противник”, второй с “Оружие”. При помощи цикла for заполнить оба массива десятью объектами со случайными значениями имён и характеристик. После чего, при помощи цикла foreach вывести их данные в консоль.

**Результат**

Было создано два массива, при помощи класса Array:

static ArrayList<Enemy> *masEnemy* = new ArrayList<>();  
static ArrayList<Weapon> *masWeapon* = new ArrayList<>();

Первый массив – это «противники», второй – «оружие». Было создано два метода – первый для заполнения массива masEnemy, второй для masWeapon, их соответственная реализация:

public static void makeEnemy() {  
 String[] names = {"Lexa", "Nika", "Katy", "Perry", "Miley", "Cyrus", "Sky", "Bloom", "Lisa", "Ksenia"};  
  
 Random rnd = new Random();  
  
 for (int i = 0; i < 10; i++) {  
 String name\_enemys = names[rnd.nextInt(10)];  
 int hp\_enemys = rnd.nextInt((100) + 1);  
 int atk\_enemys = rnd.nextInt((100) + 1);  
 int deff\_enemys = rnd.nextInt((100) + 1);  
 *masEnemy*.add(new Enemy(name\_enemys, hp\_enemys, atk\_enemys, deff\_enemys));  
 }  
}  
  
public static void makeWeapons() {  
 String[] names = {"Glock", "AK-47", "RX-22", "EP 22", "SPAS-12", "Stinger", "AT4", "RPG-7", "M9", "TMP"};  
 Random rnd = new Random();  
  
 for (int i = 0; i < 10; i++) {  
 String name\_weapon = names[rnd.nextInt(10)];  
 int atack\_weapon = rnd.nextInt((100) + 1);  
 int coast\_weapon = rnd.nextInt((100) + 1);  
 int need\_level\_weapon = rnd.nextInt((100) + 1);  
 *masWeapon*.add(new Weapon(name\_weapon, atack\_weapon, coast\_weapon, need\_level\_weapon));  
 }  
}

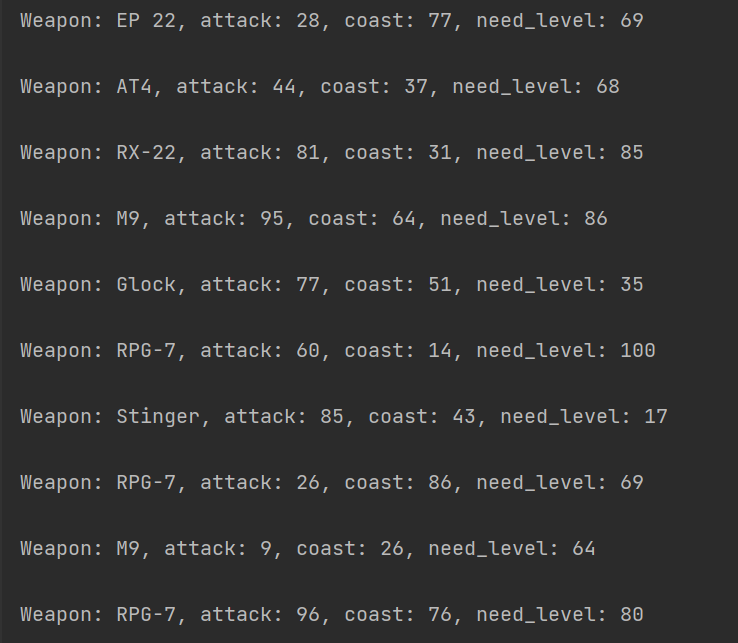
Для классов Hero и Weapon был реализован метод void show() , который выводит об объекте всю информацию. Благодаря foreach выводим наши объекты поочередно:

for(Enemy enemy: *masEnemy*) {  
 enemy.show();  
}  
  
for(Weapon weapon: *masWeapon*) {  
 weapon.show();  
}

Результат работы для Enemy:



Результат работы для Weapon:



**Приложение (на всякий случай)**

package com.company;  
  
import java.util.ArrayList;  
import java.util.Random;  
import java.util.Scanner;  
  
class GameObject {  
 protected GameObject() {}  
 protected void show() {}  
}  
  
class Hero extends GameObject {  
 private String name;  
 private int hp, exp, level, atk, deff;  
  
 public Hero(String name, int hp, int exp, int level, int atk, int deff) {  
 this.name = name;  
 this.hp = hp;  
 this.exp = exp;  
 this.level = level;  
 this.atk = atk;  
 this.deff = deff;  
 }  
  
 public Hero() {  
 name = "Hero";  
 hp = exp = level = atk = deff = 0;  
 }  
  
 public void takeAWeapon(int level\_weapon) {  
  
 if (level >= level\_weapon) {  
 System.*out*.println("Персонаж может взять оружие!\n");  
 } else {  
 System.*out*.println("Персонаж не может поднять оружие!\n");  
 }  
 }  
  
 public int getHp() {  
 return hp;  
 }  
  
 public void setHp(int hp) {  
 this.hp = hp;  
 }  
  
 public void show() {  
 System.*out*.println(  
 "\nHero: " + name + ", hp: " + hp + ", exp: " + exp + ", level: " + level + ", atk: " + atk + ", deff: " + deff  
 );  
 }  
}  
  
class Enemy extends GameObject {  
 private String name;  
 private int hp, atk, deff;  
  
 public Enemy(String name, int hp, int atk, int deff) {  
 this.name = name;  
 this.hp = hp;  
 this.atk = atk;  
 this.deff = deff;  
 }  
  
 public Enemy() {  
 name = "Enemy";  
 hp = atk = deff = 0;  
 }  
  
 public void show() {  
 System.*out*.println(  
 "\nEnemy: " + name + ", hp: " + hp + ", atk: " + atk + ", deff: " + deff  
 );  
 }  
}  
  
class Weapon extends GameObject {  
 private String name;  
 private int attack, coast, need\_level;  
  
 public Weapon(String name, int attack, int coast, int need\_level) {  
 this.name = name;  
 this.attack = attack;  
 this.coast = coast;  
 this.need\_level = need\_level;  
 }  
  
 public Weapon() {  
 name = "Weapon";  
 attack = coast = need\_level = 0;  
 }  
  
 public void show() {  
 System.*out*.println(  
 "\nWeapon: " + name + ", attack: " + attack + ", coast: " + coast + ", need\_level: " + need\_level  
 );  
 }  
}  
  
  
public class Main {  
  
 static Hero *hero* = new Hero();  
 static ArrayList<Enemy> *masEnemy* = new ArrayList<>();  
 static ArrayList<Weapon> *masWeapon* = new ArrayList<>();  
  
 public static void main(String[] args) {  
  
 *hero* = new Hero("Lexa", 77, 50, 77, 66, 40);  
 int hp = *hero*.getHp();  
 System.*out*.print("Здоровье персонажа - " + hp + '\n');  
 *hero*.takeAWeapon(77);  
  
 *makeEnemy*();  
 *makeWeapons*();  
  
 for(Enemy enemy: *masEnemy*) {  
 enemy.show();  
 }  
  
 for(Weapon weapon: *masWeapon*) {  
 weapon.show();  
 }  
 }  
  
 public static void makeEnemy() {  
 String[] names = {"Lexa", "Nika", "Katy", "Perry", "Miley", "Cyrus", "Sky", "Bloom", "Lisa", "Ksenia"};  
  
 Random rnd = new Random();  
  
 for (int i = 0; i < 10; i++) {  
 String name\_enemys = names[rnd.nextInt(10)];  
 int hp\_enemys = rnd.nextInt((100) + 1);  
 int atk\_enemys = rnd.nextInt((100) + 1);  
 int deff\_enemys = rnd.nextInt((100) + 1);  
 *masEnemy*.add(new Enemy(name\_enemys, hp\_enemys, atk\_enemys, deff\_enemys));  
 }  
 }  
  
 public static void makeWeapons() {  
 String[] names = {"Glock", "AK-47", "RX-22", "EP 22", "SPAS-12", "Stinger", "AT4", "RPG-7", "M9", "TMP"};  
 Random rnd = new Random();  
  
 for (int i = 0; i < 10; i++) {  
 String name\_weapon = names[rnd.nextInt(10)];  
 int atack\_weapon = rnd.nextInt((100) + 1);  
 int coast\_weapon = rnd.nextInt((100) + 1);  
 int need\_level\_weapon = rnd.nextInt((100) + 1);  
 *masWeapon*.add(new Weapon(name\_weapon, atack\_weapon, coast\_weapon, need\_level\_weapon));  
 }  
 }  
}