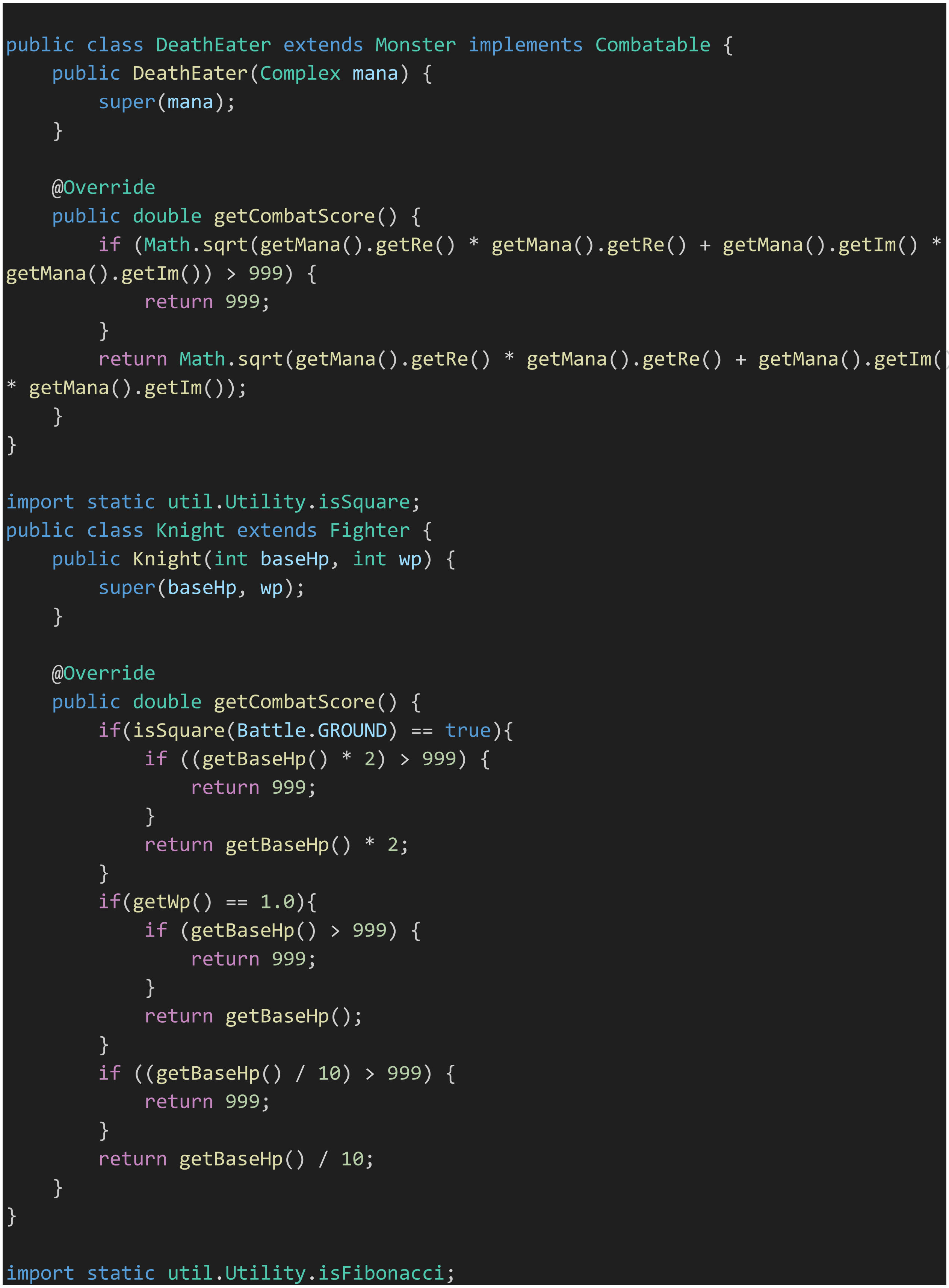


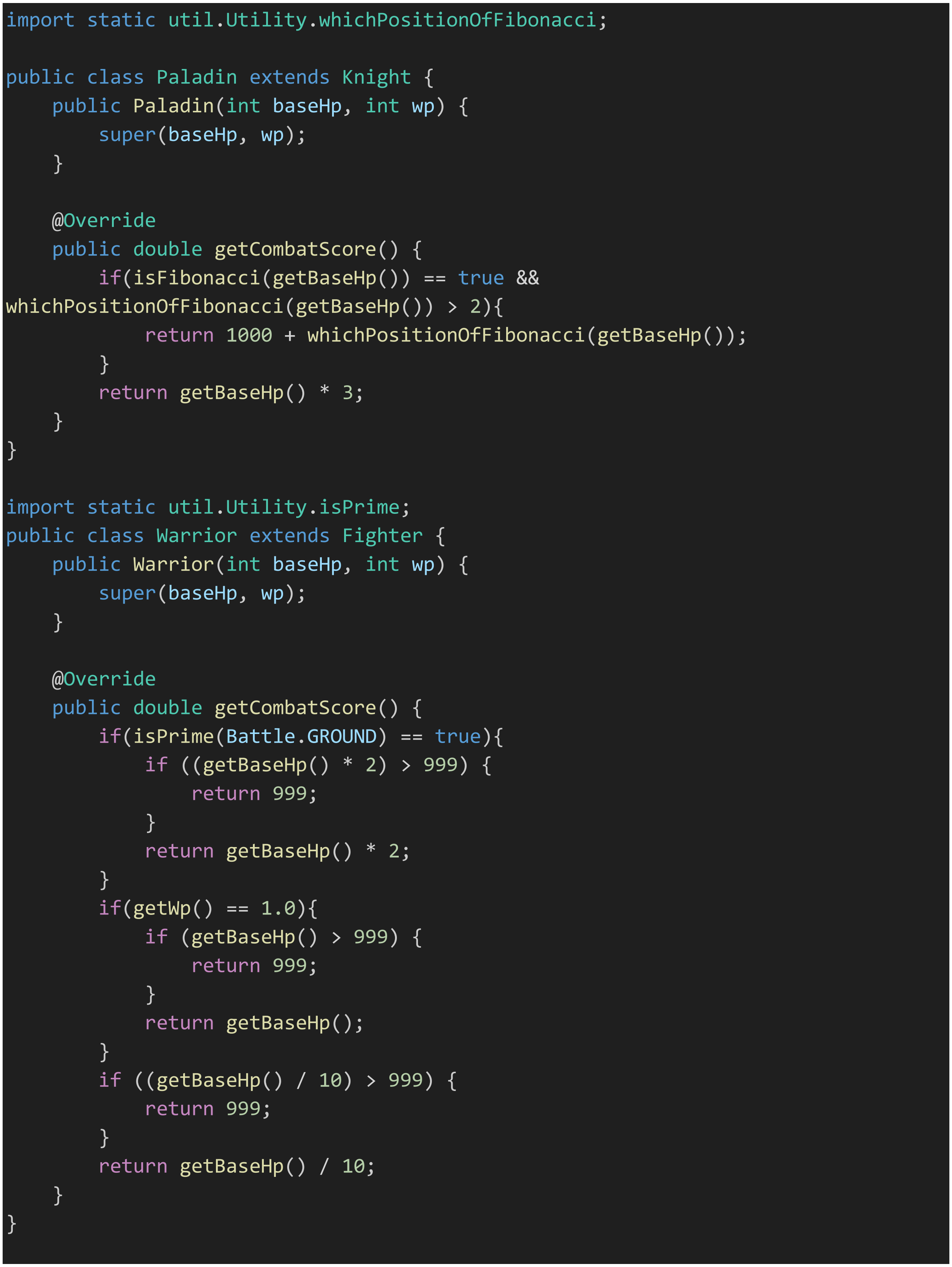
```java
public class DeathEater extends Monster implements Combatable {
    public DeathEater(Complex mana) {
        super(mana);
    }

    @Override
    public double getCombatScore() {
        if (Math.sqrt(getMana().getRe() * getMana().getRe() + getMana().getIm() *
getMana().getIm()) > 999) {
            return 999;
        }
        return Math.sqrt(getMana().getRe() * getMana().getRe() + getMana().getIm()
* getMana().getIm());
    }
}

import static util.Utility.isSquare;
public class Knight extends Fighter {
    public Knight(int baseHp, int wp) {
        super(baseHp, wp);
    }

    @Override
    public double getCombatScore() {
        if(isSquare(Battle.GROUND) == true){
            if ((getBaseHp() * 2) > 999) {
                return 999;
            }
            return getBaseHp() * 2;
        }
        if(getWp() == 1.0){
            if (getBaseHp() > 999) {
                return 999;
            }
            return getBaseHp();
        }
        if ((getBaseHp() / 10) > 999) {
            return 999;
        }
        return getBaseHp() / 10;
    }
}

import static util.Utility.isFibonacci;
```



```
import static util.Utility.whichPositionOfFibonacci;

public class Paladin extends Knight {
    public Paladin(int baseHp, int wp) {
        super(baseHp, wp);
    }

    @Override
    public double getCombatScore() {
        if(isFibonacci(getBaseHp()) == true &&
whichPositionOfFibonacci(getBaseHp()) > 2){
            return 1000 + whichPositionOfFibonacci(getBaseHp());
        }
        return getBaseHp() * 3;
    }
}

import static util.Utility.isPrime;
public class Warrior extends Fighter {
    public Warrior(int baseHp, int wp) {
        super(baseHp, wp);
    }

    @Override
    public double getCombatScore() {
        if(isPrime(Battle.GROUND) == true){
            if ((getBaseHp() * 2) > 999) {
                return 999;
            }
            return getBaseHp() * 2;
        }
        if(getWp() == 1.0){
            if (getBaseHp() > 999) {
                return 999;
            }
            return getBaseHp();
        }
        if ((getBaseHp() / 10) > 999) {
            return 999;
        }
        return getBaseHp() / 10;
    }
}
```