*ตรงจุดที่มีคะแนน 0.5 หรือ 1 คะแนน ถ้าผิด ให้ 0 เลย *

เขียนอินเตอร์เฟสตามที่สั่งในโจทย์ครบได้ 1 คะแนน ถ้าไม่ครบก็ได้ 0 เลย inferface (1 คะแนน)

คลาส FightingGame (10 คะแนน)

```
public class FightingGame {

	private Fighter player1;
	private Fighter player2;
	private Fighter winner;

	public FightingGame(Fighter player1, Fighter player2) {
		super();
		this.player1 = player1;
		this.player2 = player2;
		winner = null;
	}

	public void gameLoop() {
		while (true) {
			player1.move();
			player2.move();
			updateGameStatus();
			if (winner == player1) {
				System.out.println("Player 1 wins");
				break;
			}
			if (winner == player2) {
				System.out.println("Player 2 wins");
				break;
			}
			if (winner != null) {
				System.out.println("Double KO");
				break;
			}

		}
	}

	public void updateGameStatus() {
		if (player1.getLifePercent() <= 0.0 && player2.getLifePercent() <= 0.0) {
			winner = new KungFuFighter();
			return;
		} else if (player2.getLifePercent() <= 0.0) {
			winner = player1;
			return;
		} else if (player1.getLifePercent() <= 0.0) {
			winner = player2;
			return;
		}
	}

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Fighter firstPlayer = new FinalFantasyFighter();
		Fighter secondPlayer = new KungFuFighter();
		firstPlayer.setOpponent(secondPlayer);
		secondPlayer.setOpponent(firstPlayer);

		FightingGame g = new FightingGame(firstPlayer, secondPlayer);
		g.gameLoop();
	}
```

1 คะแนน ถ้าผิดนิดเดียว 0 เลย ตัวแปรจะต้องเป็น private และเป็นชนิด Fighter ทั้งสามตัว

ชนิดพารามิเตอร์ต้องเป็น Fighter และต้องทำถูกตามลูกศรทั้งหมด จึงจะได้ 2 คะแนน ถ้ามีผิด หักจุดละ 1 คะแนน

ลูปเป็นลูปตลอดเวลา ได้ 1 คะแนน

เรียก move ของ player ทั้งสองคน 1 คะแนน

พิมพ์ข้อความถูกต้องตามผู้ชนะ

1 คะแนน

หาว่าใครเป็นผู้ชนะ 1 คะแนน

อย่าลืมว่า พลังชีวิต อ่านได้แต่ getLifePercent เท่านั้น จะใช้ getLife ของคลาสอื่นไม่ได้

สร้าง Fighter สองชนิดตามโจทย์ 1 คะแนน

ต้องมี constructor ที่มีพารามิเตอร์ตามที่สร้างตรงนี้จริงนะ

ให้แต่ละตัวรับรู้ว่าคู่ต่อสู้เป็นใคร 1 คะแนน

New FightingGame แล้วเรียก gameLoop

1 คะแนน

}

คลาส KungFuMan เป็นคลาสต้นฉบับ ต้องปล่อยไว้เฉยๆโดยไม่เปลี่ยนอะไรเลย ถ้ามีการเปลี่ยน ต้องไม่ให้คะแนน KungFuFighter ทั้งคลาส

```
public class KungFuMan {
	/**
	 * An opponent of this KungFuMan.
	 */
	private Object opponent;

	/**
	 * Life point.
	 */
	private int life;
	private int maxLife = 100;

	/**
	 * Strength, combining with defence must not exceed 100.
	 */
	private int strength;

	/**
	 * Defence, combining with strength must not exceed 100.
	 */
	private int defence;

	/**
	 * Heavier move requires longer time to recover. While Recovering, the
	 * character cannot attack.
	 */
	private int recoverTimeFromOwnMove;

	public KungFuMan(Object opponent) {
		super();
		this.opponent = opponent;
		life = maxLife;
		strength = 50; //strength + defence == 100  always
		defence = 50;
		recoverTimeFromOwnMove = 0;
	}

	/**
	 * Get a reference to the current opponent.
	 *
	 * @return opponent.
	 */
	public Object getOpponent() {
		return opponent;
	}

	/**
	 * Assign a given KungFuMan to be the current opponent.
	 *
	 * @param opponent
	 */
	public void setOpponent(Object opponent) {
		this.opponent = opponent;
	}

	public int getLife() {
		return life;
	}

	public void setLife(int life) {
		this.life = life;
	}

	public int getMaxLife() {
		return maxLife;
	}

	public void setMaxLife(int life) {
		this.maxLife = life;
```

```
}


public int getStrength() {
        return strength;
}

public void setStrength(int strength) {
        this.strength = strength;
}

public int getDefence() {
        return defence;
}

public void setDefence(int defence) {
        this.defence = defence;
}

public int getRecoverTimeFromOwnMove() {
        return recoverTimeFromOwnMove;
}

public void setRecoverTimeFromOwnMove(int recoverTimeFromOwnMove) {
        this.recoverTimeFromOwnMove = recoverTimeFromOwnMove;
}

/**
 * If not recovering, always attack with attack type 3.
 */
public void move() {
        if (recoverTimeFromOwnMove == 0)
                attack(3);
        else
                recoverTimeFromOwnMove -= 1;
}

/**
 * Reduce the life point after being attacked.
 *
 * @param attackLevel
 *            indicates the level of attack: 1 -> light 2-> medium 3 ->
 *            heavy
 */
public void attacked(int attackLevel) {
        life = life - (attackLevel * ((KungFuMan) opponent).strength / defence);
}

/**
 * Attack opponent with given type of attack (from 1 to 7). Resulting in
 * opponent calling method attacked().
 *
 * @param attackType
 *            the moves that this possesses.
 */
public void attack(int attackType) {
        recoverTimeFromOwnMove = attackType;

        switch (attackType) {
        case 1:
                ((KungFuMan) opponent).attacked(1);
                System.out.println(" straight light punch.");
                break;

        case 2:
                ((KungFuMan) opponent).attacked(1);
                System.out.println(" straight light kick.");
                break;

        case 3:
                ((KungFuMan) opponent).attacked(2);
                System.out.println(" straight medium punch.");
                break;

        case 4:
                ((KungFuMan) opponent).attacked(2);
                System.out.println(" straight medium kick.");
```

```
                        break;

                case 5:
                        ((KungFuMan) opponent).attacked(3);
                        System.out.println(" straight heavy punch.");
                        break;

                case 6:
                        ((KungFuMan) opponent).attacked(3);
                        System.out.println(" straight heavy kick.");
                        break;

                case 7:
                        ((KungFuMan) opponent).attacked(3);
                        System.out.println(" Roundhouse heavy kick.");
                        break;

                default:
                        System.out.println(" Move failed!");
                        break;
                }
        }

}
```

คลาส KungFuFighter (11 คะแนน)

Extends กับ implements ถูกตามนี้ ได้ 1 คะแนน

```
public class KungFuFighter extends KungFuMan implements Fighter {
        public KungFuFighter() {
                super(null);
                // TODO Auto-generated constructor stub
        }

        public KungFuFighter(Object opponent) {
                super(opponent);
                // TODO Auto-generated constructor stub
        }

        public double getDefencePercent() {
                // TODO Auto-generated method stub
                return (getDefence() * 100.0) / (getDefence() + getStrength());
        }

        public double getLifePercent() {
                // TODO Auto-generated method stub
                return (double) (getLife() * 100.0)/getMaxLife();
        }

        public double getStrengthPercent() {
                // TODO Auto-generated method stub
                return (getStrength() * 100.0) / (getDefence() + getStrength());
        }

        public int getRecoverTime() {
                return getRecoverTimeFromOwnMove();
        }

        public void setRecoverTime(int time){
                setRecoverTimeFromOwnMove(time);
        }

        public void move() {
                Fighter f = (Fighter) getOpponent();

                if (getRecoverTimeFromOwnMove() == 0 && f.getLifePercent() <= 25)
                        attack(3);

                else if (getRecoverTimeFromOwnMove() == 0 && f.getLifePercent() <= 65) {
                        attack(2);
                } else if (getRecoverTimeFromOwnMove() == 0
                                && f.getLifePercent() > 65) {
                        attack(1);
                }
```

ต้องให้ค่าตัวเลข set เหมือนกับคอนสตรัคเตอร์ของซุปเปอร์คลาส ได้ 1 คะแนน (แสดงคอนสตรัคเตอร์ที่ถูกตามนี้แค่อันเดียวก็พอ)

1 คะแนน ใช้ 100 ก็ได้ไม่ต้องทศนิยม แต่ต้องคูณก่อนเท่านั้น

ตามนี้ 1 คะแนน

ตามนี้ 1 คะแนน ไม่ต้องใช้ทศนิยม แต่ต้องคูณก่อน

ตามนี้ 1 คะแนน มัน access data โดยตรงไม่ได้ ก็ต้องเรียกเมธอดแบบนี้

ตามนี้ 1 คะแนน มัน access data โดยตรงไม่ได้ ก็ต้องเรียกเมธอดแบบนี้

ถ้ามีการสร้างตัวแปรขึ้นมาซ้ำกับใน KungFuMan ให้ถือว่า การใช้งานตัวแปรที่ประกาศใหม่นั้น ผิด ทุกที่

ตรงนี้ถูกหมดได้ 2 คะแนน ถ้ามีผิดหักจุดละ 1 คะแนน

การ get กับ set recover time ใช้เมธอดของ superclass หรือคลาสนี้ก็ได้

```
                else
                        setRecoverTimeFromOwnMove(getRecoverTimeFromOwnMove() - 1);

        }


        public void attack(int attackType) {
                setRecoverTimeFromOwnMove(attackType);
                Fighter enemy = (Fighter) getOpponent();

                switch (attackType) {
                case 1:
                        enemy.attacked(1);
                        System.out.println(" straight light punch.");
                        break;

                case 2:
                        enemy.attacked(1);
                        System.out.println(" straight light kick.");
                        break;

                case 3:
                        enemy.attacked(2);
                        System.out.println(" straight medium punch.");
                        break;

                case 4:
                        enemy.attacked(2);
                        System.out.println(" straight medium kick.");
                        break;

                case 5:
                        enemy.attacked(3);
                        System.out.println(" straight heavy punch.");
                        break;

                case 6:
                        enemy.attacked(3);
                        System.out.println(" straight heavy kick.");
                        break;

                case 7:
                        enemy.attacked(3);
                        System.out.println(" Roundhouse heavy kick.");
                        break;

                default:
                        System.out.println(" Move failed!");
                        break;
                }
        }

        public void attacked(int attackLevel) {
                double life = getLifePercent();
                double defence = getDefencePercent();
                double opponentStrength = ((Fighter)getOpponent()).getStrengthPercent();

                life = life - (attackLevel * opponentStrength / defence);
                setLife((int)(life * getMaxLife()/100.0)  ) ;

}
```

เปลี่ยนศัตรูให้เป็น Fighter 1 คะแนน

โค้ดเกือบเหมือนเดิมแต่จะต้องไม่มีศัตรูที่เป็น KungFuMan อีก ถ้ามี KungFuMan (เท่านั้น) ให้ -1 คะแนนไป

เหมือนในซุปเปอร์คลาส แต่ว่าค่า strength, defence ต้องเป็นค่าร้อยละ ทำถูกได้ 1 คะแนน

คลาส FinalFantasyFighter(10 คะแนน)

```
public class FinalFantasyFighter implements Fighter {
        private static final int FF_MAXSTAT = 255;
        private Fighter opponent;
        private int timeToRecover;
        private int life;
        private int strength;

        public FinalFantasyFighter() {
```

Implements ถูกต้อง 1 คะแนน

ตัวแปรใส่ครบตามนี้ 1 คะแนน ต้องเป็น private หมดด้วย

```
        this.opponent = null;
        this.timeToRecover = 0;
        this.life = FF_MAXSTAT;
        this.strength = 130;
}

public FinalFantasyFighter(Fighter opponent) {

        this.opponent = opponent;
        this.timeToRecover = 0;
        this.life = FF_MAXSTAT;
        this.strength = 130;
}

public Fighter getOpponent() {
        return opponent;
}

public void setOpponent(Object opponent) {
        this.opponent = (Fighter) opponent;
}

public int getLife() {
        return life;
}

public void setLife(int life) {
        this.life = life;
}

public int getStrength() {
        return strength;
}

public void setStrength(int strength) {
        this.strength = strength;
}

@Override
public int getRecoverTime() {
        // TODO Auto-generated method stub
        return timeToRecover;
}

@Override
public void setRecoverTime(int time) {
        // TODO Auto-generated method stub
        timeToRecover = time;
}


@Override
public void attack(int attackType) {
        setRecoverTime(attackType);
        Fighter enemy = getOpponent();

        switch (attackType) {
        case 1:
                enemy.attacked(1);
                System.out.println("light lunge.");
                break;

        case 2:
                enemy.attacked(2);
                System.out.println(" medium lunge.");
                break;

        case 3:
                enemy.attacked(3);
                System.out.println(" Heavy lunge.");
                break;

        default:
                System.out.println(" FF fighter move failed!");
                break;
        }
```

ต้องมีคอนสตรัคเตอร์ที่เซ็ตค่าเริ่มต้นได้หมด 1 คะแนน

ตรงนี้ถูก ถึงจะ type ไม่ตรงกับใน interface แต่จาว่ายอม ดังนั้นจะถือเป็น get ไปด้วย เป็นเมธอดที่ implement จาก interface ไปด้วย

ถ้ามีตรงนี้ครบสี่คู่ ได้ 1 คะแนน ไม่งั้นก็ไม่ได้คะแนนเลย ชื่อเมธอดไม่เหมือนในอินเตอร์เฟสก็ได้ แต่ต้องจัดการครบทุกตัวแปรที่คลาสนี้มี

ถ้ามีตรงนี้ครบ 1 คะแนน ไม่งั้นก็ไม่ได้คะแนนเลย

```
	}

	@Override
	public void attacked(int attackLevel) {
		double life = getLifePercent();
		double defence = getDefencePercent();
		double opponentStrength = ((Fighter) getOpponent())
				.getStrengthPercent();

		life = life - (attackLevel * opponentStrength / defence);
		setLife((int) (life * FF_MAXSTAT / 100));
	}


	@Override
	public double getDefencePercent() {
		// TODO Auto-generated method stub
		double defence = FF_MAXSTAT - strength;
		double defencePercent = defence * 100.0 / FF_MAXSTAT;
		return defencePercent;

	}

	@Override
	public double getLifePercent() {
		// TODO Auto-generated method stub
		return (double)life * 100.0 / FF_MAXSTAT;
	}

	@Override
	public double getStrengthPercent() {
		// TODO Auto-generated method stub
		double strengthPercent = strength * 100.0 / FF_MAXSTAT;
		return strengthPercent;
	}

	@Override
	public void move() {
		// TODO Auto-generated method stub
		Fighter f = getOpponent();

		if (getRecoverTime() == 0 && f.getLifePercent() <= 25)
			attack(1);

		else if (getRecoverTime() == 0 && f.getLifePercent() <= 65) {
			attack(2);
		} else if (getRecoverTime() == 0 && f.getLifePercent() > 65) {
			attack(3);
		}

		else
			setRecoverTime(getRecoverTime() - 1);
	}

}
```

ครบนี่ได้ 1 คะแนน แต่ว่า setLife ต้องปัดเลขเป็น int ตอนท้ายเลยนะ ห้ามปัดแต่เนิ่นๆ

ต้องทำให้ข้อมูลอยู่ในรูปแบบ double นะ

1 คะแนน

ต้องทำให้ข้อมูลอยู่ในรูปแบบ double นะ

1 คะแนน

ต้องทำให้ข้อมูลอยู่ในรูปแบบ double นะ

1 คะแนน

เขียนตามนี้ครบ ได้ 1 คะแนน

ผลการรันใน text file (2 คะแนน)

```
Heavy lunge.
 straight light punch.
 straight light punch.
```

ซ้ำไปหลายที

0.5 คะแนน

```
medium lunge.
 straight light punch.
 straight light punch.
 medium lunge.
 straight light punch.
```

0.5 คะแนน

ซ้ำไปหลายที

```
Medium lunge.
 straight light kick.
```

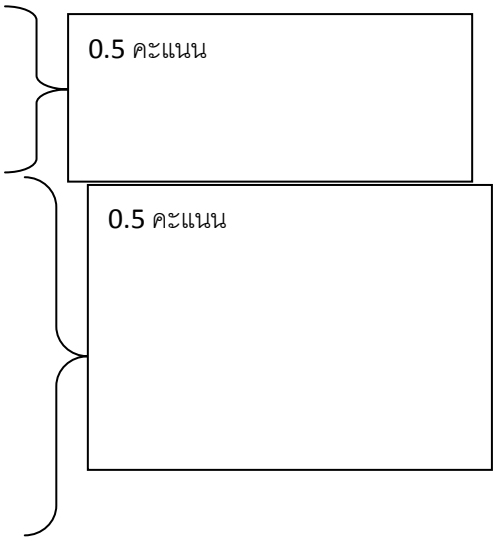
ซ้ำไปหลายที

```
light lunge.
 straight light kick.
light lunge.
 straight light kick.
light lunge.
```

ซ้ำไปหลายที

แล้วจบด้วย

```
light lunge.
 straight light kick.
Player 1 wins
```



รวม 34 คะแนน