

Java 課題レポート - 第9回

学籍番号: 24024 名前: 白石鷹也 授業日: 11/08

過去に作成したプログラムの変更

練習12で作成したHeroクラスの変数の可視性を以下のように変更しました。

```
public class Practice12Hero {  
    public int hp;  
    protected String name;  
    private Practice12Sword sword;
```

課題006

スタブ

コンパイルエラーが出ないように、以下のスタブを作成しました。

```
package period9.Kadai6;  
  
public class Hero {  
    public int hp;  
  
    public Hero() {  
        this.hp = 100;  
    }  
}
```

問題1

ソースコード

```
package period9.Kadai6;
public class PoisonMatango extends Matango {
    int poisonCount = 5; // 毒攻撃の残り回数

    public PoisonMatango(char suffix) {
        super(suffix);
    }

    @Override
    public void attack(Hero h) {
        // 通常の攻撃
        super.attack(h);

        // 毒攻撃が可能な場合
        if (poisonCount > 0) {
            System.out.println("さらに毒の胞子をばらまいた!");
            int poisonDamage = h.hp / 5; // 勇者のHPの1/5のダメージ
            h.hp -= poisonDamage;
            System.out.println("ポイントのダメージ! " + poisonDamage + " のダメージを受けた");
            poisonCount--; // 毒攻撃の残り回数を減らす
        }
    }
}
```

練習13

ソースコード

```
package period9;
import java.util.Scanner;
import period8.Practice12Sword;
public class Practice13 {
    public static void main(String[] args) {
        try(Scanner scanner = new Scanner(System.in)){
            // プレイヤーに名前を入力してもらう
            System.out.print("勇者の名前を付けて下さい: ");
            String heroName = scanner.nextLine();
            System.out.print("剣の名前をつけてください: ");
            String swordName = scanner.nextLine();

            // スーパー勇者を作成し、剣を装備させる
            Practice12Sword sword = new Practice12Sword(swordName);
            Practice13SuperHero hero = new Practice13SuperHero(heroName,
sword);

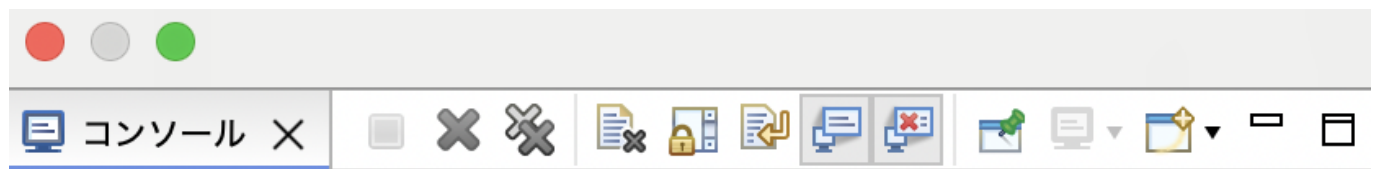
            System.out.println("勇者は" + heroName + "と名付けられた");
            System.out.println("剣は" + swordName + "と名付けられた");

            while (true) {
                System.out.println("勇者" + hero.getName() + "は冒険に出かけ
た。 指示をしてください。");
                System.out.println("1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:
着陸する 9:終了");
                int command = scanner.nextInt();

                switch (command) {
                    case 1:
                        hero.fight();
                        break;
                    case 2:
                        hero.runAway();
                        break;
                    case 3:
                        hero.sleep();
                        break;
                    case 4:
                        hero.swordFight();
                        break;
                    case 5:
                        hero.fly();
                        break;
                    case 6:
                        hero.land();
                        break;
                    case 9:
                        hero.endAdventure();
                        return;
                }
            }
        }
    }
}
```

```
        default:
            hero.pass();
        }
    }
}
```

実行結果



<終了> Practice13 [Java アプリケーション] /Users/pality/.sdkman/candidates/java/

勇者の名前を付けて下さい: テスト

剣の名前をつけてください: テストソード

勇者はテストと名付けられた

剣はテストソードと名付けられた

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

1

勇者テストは戦った。 hp:80

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

2

勇者テストは逃げた。

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

3

勇者テストは眠って、hpを回復した

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

4

勇者テストはテストソードを使って戦った。 相手に20のダメージを与えた

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

5

勇者テストは空を飛んだ。

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

5

勇者テストは既に空を飛んでいる。

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

6

勇者テストは着陸した。

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

6

勇者テストは空を飛んでいないので着陸できない。

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

8

勇者テストはスルーした

勇者テストは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 9:終了

9

勇者テストの冒険は終了した

練習13

ソースコード

SuperHeroクラスです。

```
package period9;

import period8.Practice12Hero;
import period8.Practice12Sword;

public class Practice13SuperHero extends Practice12Hero {
    private boolean isFlying = false; // 飛行状態を管理するフラグ

    // コンストラクタ：名前と剣を受け取り、親クラスのコンストラクタを呼び出す
    public Practice13SuperHero(String name, Practice12Sword sword) {
        super(name, sword);
    }

    // 飛ぶメソッド
    public void fly() {
        if (!isFlying) {
            System.out.println("勇者" + this.getName() + "は空を飛んだ。");
            isFlying = true;
        } else {
            System.out.println("勇者" + this.getName() + "は既に空を飛んでいる。");
        }
    }

    // 着陸するメソッド
    public void land() {
        if (isFlying) {
            System.out.println("勇者" + this.getName() + "は着陸した。");
            isFlying = false;
        } else {
            System.out.println("勇者" + this.getName() + "は空を飛んでいないので着陸できない。");
        }
    }

    // 名前を取得するメソッド
    public String getName() {
        return this.name;
    }
}
```

練習14

ソースコード

```
package period9;

import java.util.Scanner;

import period8.Practice12Sword;

public class Practice14 {
    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {
            // プレイヤーに名前を入力してもらう
            System.out.print("勇者の名前を付けて下さい: ");
            String heroName = scanner.nextLine();
            System.out.print("剣の名前をつけてください: ");
            String swordName = scanner.nextLine();

            // スーパー勇者を作成し、剣を装備させる
            Practice12Sword sword = new Practice12Sword(swordName);
            Practice14SuperHero hero = new Practice14SuperHero(heroName,
sword);

            System.out.println("勇者は" + heroName + "と名付けられた");
            System.out.println("剣は" + swordName + "と名付けられた");

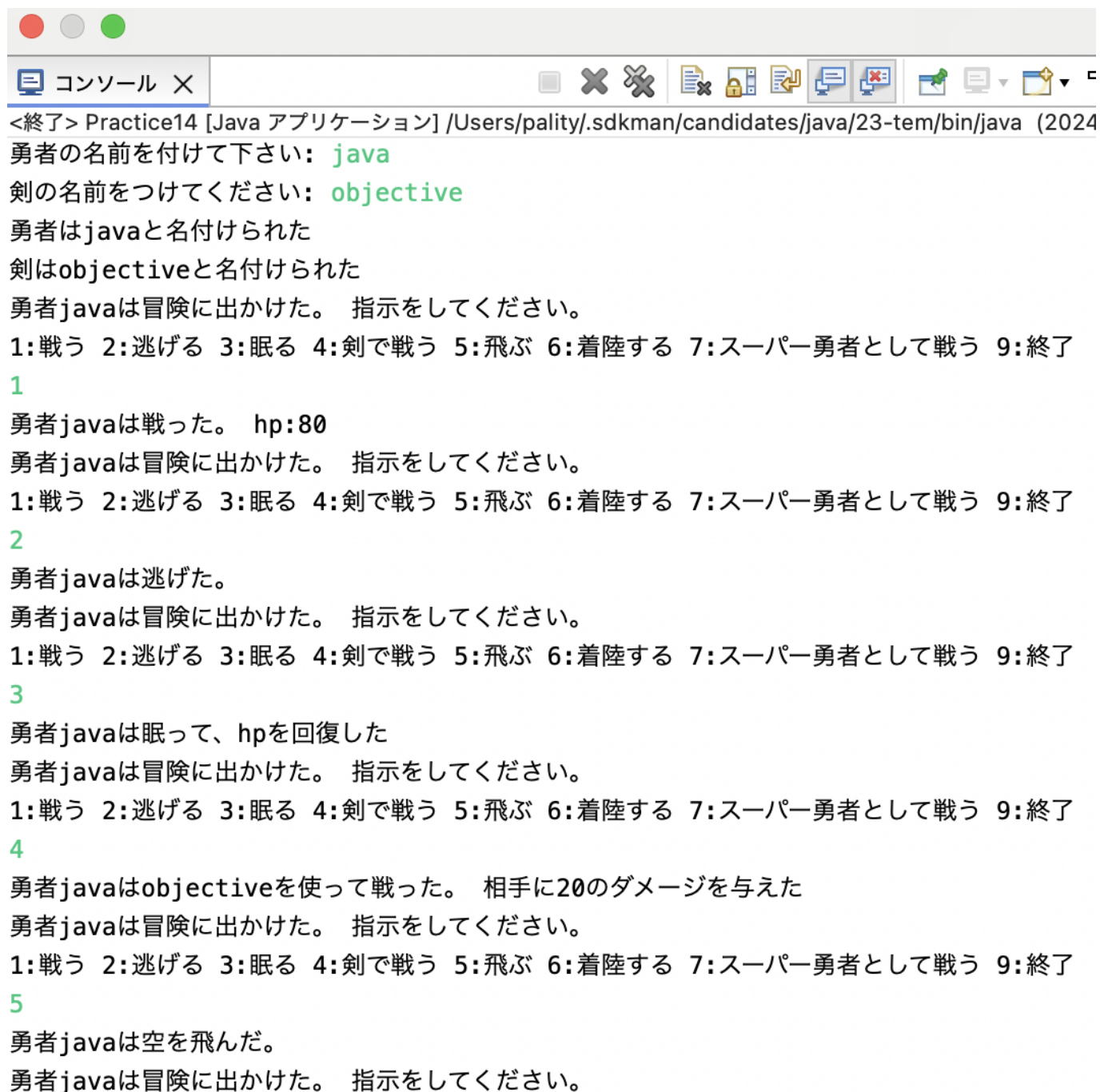
            while (true) {
                System.out.println("勇者" + hero.getName() + "は冒険に出かけ
た。 指示をしてください。");
                System.out.println("1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:
着陸する 7:スーパー勇者として戦う 9:終了");
                int command = scanner.nextInt();

                switch (command) {
                    case 1:
                        hero.normalFight(); // 親クラスの戦うメソッドを呼ぶ
                        break;
                    case 2:
                        hero.runAway();
                        break;
                    case 3:
                        hero.sleep();
                        break;
                    case 4:
                        hero.swordFight();
                        break;
                    case 5:
                        hero.fly();
                        break;
                    case 6:
                        hero.land();
                        break;
```



```
        case 7:
            hero.fight(); // 子クラスのオーバーライドされた戦うメソッ
            break;
        case 9:
            hero.endAdventure();
            return;
        default:
            hero.pass();
    }
}
}
```

実行結果



```
<終了> Practice14 [Java アプリケーション] /Users/pality/.sdkman/candidates/java/23-tem/bin/java (2024
勇者の名前を付けて下さい: java
剣の名前をつけてください: objective
勇者はjavaと名付けられた
剣はobjectiveと名付けられた
勇者javaは冒険に出かけた。 指示をしてください。
1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了
1
勇者javaは戦った。 hp:80
勇者javaは冒険に出かけた。 指示をしてください。
1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了
2
勇者javaは逃げた。
勇者javaは冒険に出かけた。 指示をしてください。
1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了
3
勇者javaは眠って、hpを回復した
勇者javaは冒険に出かけた。 指示をしてください。
1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了
4
勇者javaはobjectiveを使って戦った。 相手に20のダメージを与えた
勇者javaは冒険に出かけた。 指示をしてください。
1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了
5
勇者javaは空を飛んだ。
勇者javaは冒険に出かけた。 指示をしてください。
```


1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了

5

勇者javaは既に空を飛んでいる。

勇者javaは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了

6

勇者javaは着陸した。

勇者javaは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了

6

勇者javaは空を飛んでいないので着陸できない。

勇者javaは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了

7

勇者javaは戦った。 hpは、減らなかった。

勇者javaは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了

8

勇者javaはスルーした

勇者javaは冒険に出かけた。 指示をしてください。

1:戦う 2:逃げる 3:眠る 4:剣で戦う 5:飛ぶ 6:着陸する 7:スーパー勇者として戦う 9:終了

9

勇者javaの冒険は終了した

練習14

ソースコード

```
package period9;
import period8.Practice12Hero;
import period8.Practice12Sword;
public class Practice14SuperHero extends Practice12Hero {
    private boolean isFlying = false; // 飛行状態を管理するフラグ

    // コンストラクタ：名前と剣を受け取り、親クラスのコンストラクタを呼び出す
    public Practice14SuperHero(String name, Practice12Sword sword) {
        super(name, sword);
    }

    // 通常の「戦う」メソッド（親クラスのメソッドを呼ぶ）
    public void normalFight() {
        super.fight();
    }

    // スーパー勇者として戦うメソッド（オーバーライド）
    @Override
    public void fight() {
        System.out.println("勇者" + this.getName() + "は戦った。 hpは、減らなかつた。");
    }

    // 飛ぶメソッド
    public void fly() {
        if (!isFlying) {
            System.out.println("勇者" + this.getName() + "は空を飛んだ。");
            isFlying = true;
        } else {
            System.out.println("勇者" + this.getName() + "は既に空を飛んでいる。");
        }
    }

    // 着陸するメソッド
    public void land() {
        if (isFlying) {
            System.out.println("勇者" + this.getName() + "は着陸した。");
            isFlying = false;
        } else {
            System.out.println("勇者" + this.getName() + "は空を飛んでいないので着陸できない。");
        }
    }

    // 名前を取得するメソッド
    public String getName() {
        return this.name;
    }
}
```

```
    }  
}
```