情報学群実験第1 第12回授業 演習課題

平成 29 年 5 月 26 日

1　課題 1

下記の実行結果が得られるように抽象クラス Character を継承した未完成の Hero クラスと Enemy クラスを完成させなさい．各クラスの仕様は表 1，2，3 の通りである．Lesson11_1.java，Character.java は変更しないものとする．

表 1: Character クラスの仕様

フィールド	説明
private int hp	HP の値を格納するフィールド変数
private String name	キャラクターの名前を格納するフィールド変数
コンストラクタ	
public Character(String name, int hp)	フィールド変数の値を初期化するコンストラクタ
メソッド	
public void damege(int atk)	atk の値のダメージを受ける処理を行うメソッド
public String getName()	name の値を返すメソッド
public int getHp()	hp の値を返すメソッド
public abstract void printName()	キャラクターの名前を表示する抽象メソッド

表 2: Hero クラスの仕様

フィールド	説明
private int atk	atk の値を格納するフィールド変数
コンストラクタ	
public Hero(String name, int hp, int atk)	それぞれの値をセットするコンストラクタ
メソッド	
public int getAtk()	atk の値を返すメソッド
public void attack(Enemy enemy)	enemy に atk 分のダメージを与え出力する
public void printName()	キャラクターの名前を表示するメソッド. 親クラスの抽象メソッドをオーバーライドする.

表 3: Enemy クラスの仕様

フィールド	説明
private int atk	atk の値を格納するフィールド変数
コンストラクタ	
public Enemy(String name, int hp, int atk)	それぞれの値をセットするコンストラクタ
メソッド	
public int getAtk()	atk の値を返すメソッド
public void attack(Hero hero)	hero に atk 分のダメージを与え出力する
public void printName()	キャラクターの名前を表示するメソッド. 親クラスの抽象メソッドをオーバーライドする.

期待される入力と出力

```
$ javac Lesson11_1.java
$ java Lesson11_1
プレイヤーのキャラクターは勇者だ！
敵のキャラクターはスライムだ！
勇者の攻撃！
勇者はスライムに 16 のダメージを与えた！
```

2 課題 2

インタフェース CharInfo は，キャラクターのステータスを表示するためのメソッドを実装させることを目的としたインタフェースである．この CharInfo インタフェースを下記の実行結果が得られるように CharInfo からメソッドの仕様を読み取り，Hero クラスと Enemy クラスにメソッドを実装しなさい．また，ダメージを受けたキャラクターの残り体力を表示させるように Hero クラスと Enemy クラスを適宜修正しなさい．

※コンパイル時に下記のようなエラーメッセージが出た際は，インタフェースを実装するときのアクセス修飾子に注意する

エラーメッセージ

```
$ javac Lesson11_2.java
./Hero.java:18: エラー: Hero の showStatus() は CharInfo の showStatus() を
実装できません
    void showStatus(){
         ^
  (public) より弱いアクセス権限を割り当てようとしました
エラー 1 個
```

期待される入力と出力

```
$ javac Lesson11_2.java
$ java Lesson11_2
プレイヤーのキャラクターは勇者だ！
勇者のステータス
HP:100
ATK:16
敵のキャラクターはスライムだ！
スライムのステータス
HP:50
ATK:5
勇者の攻撃！
勇者はスライムに 16 のダメージを与えた！
スライムの残り HP は 34
```

3　課題 3

Critical インタフェースはヒーローの攻撃のクリティカルヒット時の攻撃力を計算するメソッドを実装させるためのインタフェースである．Critical を下記の条件に従って定義し，Hero クラスにクリティカルヒット時の攻撃を行うメソッドを追加せよ．

- クリティカル時の atk 倍率を定義する int 型定数 CRITICAL．値は 2 で定義すること．
- 引数に int atk を取り，クリティカルヒットした時の ATK を返す calcCritical メソッド．

期待される入力と出力

```
$ javac Lesson11_3.java
$ java Lesson11_3
プレイヤーのキャラクターは勇者だ！
勇者のステータス
HP:100
ATK:16
敵のキャラクターはスライムだ！
スライムのステータス
HP:50
ATK:5
勇者の攻撃！
クリティカルヒット !!
勇者はスライムに 32 のダメージを与えた！
スライムの残り HP は 18
```

4　課題 4

Slime クラスと Dragon クラスはそれぞれスライムとドラゴンの情報を管理するクラスである．この 2 つのクラスは共通するメソッドやフィールドが多く，別々に定義するのは無駄が多い．そこで Slime クラスと Dragon クラスのスーパークラスとなる抽象クラス Monster を定義することにした．Monster クラスについて与えられた Slime クラスと Dragon クラスを基に共通部分を抽出して Monster クラスを定義しなさい．Slime クラスと Dragon クラスについては適宜書き換えて良い．

同じメソッドやフィールドはスーパークラスで実装し，同じ用途や名前だが処理が異なるものは抽象メソッドなどにすること．

期待される入力と出力

```
$ javac Lesson11_4.java
$ java Lesson11_4
メタルスライムが現れた！
Name:メタルスライム
Type:Slime
Level:14
HP:100
ブラックドラゴンが現れた！
Name:ブラックドラゴン
Type:Dragon
Level:30
HP:300
MP:25
ブラックドラゴンの攻撃
メタルスライムは逃げ出した
```

5　課題 5

Clonable インタフェースは，オブジェクトの正当なコピーが生成できる clone メソッドが使えることを示すためのインタフェースである．Slime クラスに clone メソッドを実装し，スライムが分裂できるように実装しなさい．

期待される入力と出力

```
$ javac Lesson11_5.java
$ java Lesson11_5
スライム A が現れた！
Name:スライム A
Type:Slime
Level:14
HP:100
スライム A が分裂し，スライム B が誕生した！
Name:スライム B
Type:Slime
Level:14
HP:100
```