

■問題1(計95点・部分点有り)

ロールプレイングゲーム(RPG)のソフトウェアを考える。定義するクラス、インタフェイスは以下の通りである。設問(1)~(3)に答えよ。

- (a)RPGに登場するキャラクター般を表す public な抽象クラス RPGCharacter。ファイル RPGCharacter.java で定義され、パッケージ com.tuis.rpg.character に所属する。このクラスは次のメンバを持つ。
- ・生成された RPGCharacter 型オブジェクトの総数を表す int 型の private な static フィールド num(初期値は0)
  - ・体力を表す private な int 型フィールド hp
  - ・魔法力を表す private な int 型フィールド mp
  - ・num の public で static なゲッタ int getNum( )
  - ・hp の public なゲッタ int getHp( ), hp の public なセッタ void setHp( int hp )
  - ・mp の public なゲッタ int getMp( ), mp の public なセッタ void setMp( int mp )
  - ・仮引数 c に渡された他の RPGCharacter 型オブジェクトに攻撃をする public な抽象メソッド void attack( RPGCharacter c )
  - ・仮引数 damage に渡されたダメージ量を軽減して、軽減しきれなかったダメージの分だけ自分の hp を減らすという防衛行為を表した public な抽象メソッド void defence( int damage )
  - ・フィールド hp, mp を初期化し、num の値を1増やす public なコンストラクタ RPGCharacter( int hp, int mp )
- (b)「描画が可能である」という性質を表す public なインタフェイス Drawable。ファイル Drawable.java で定義され、パッケージ com.tuis.rpg.system に所属する。このインタフェイス Drawable は次のメンバを持つ。
- ・「そのオブジェクトを描画する」という処理を表す抽象メソッド void draw( )
- (c)「騎士」を表す public なクラス Knight (RPGCharacter のサブクラスで、Drawable を実装する)。ファイル Knight.java で定義され、パッケージ com.tuis.rpg.character に所属する。このクラスは次のメンバを持つ。
- ・RPGCharacter のコンストラクタを呼び出して、フィールド hp, mp を初期化する public なコンストラクタ Knight( int hp, int mp )
  - ・自分の hp の 1/5 だけ攻撃相手(仮引数 c )の defence メソッドに渡すようにオーバーライドした public な attack メソッド。
  - ・受けたダメージ量(仮引数 damage )から自分の hp の 1/5 だけ引いた値が0以上の場合には、その値の分だけ自分の hp を減らすようにオーバーライドした public な defence メソッド。
  - ・騎士を表す文字列 "騎士" を System.out.println メソッドで表示するようにオーバーライドした public な draw メソッド
- (d)「魔法使い」を表す public なクラス Mage (RPGCharacter のサブクラスで、Drawable を実装する)。ファイル Mage.java で定義され、パッケージ com.tuis.rpg.character に所属する。このクラスは次のメンバを持つ。
- 次の様なメンバを持つ。
- ・RPGCharacter のコンストラクタを呼び出して、フィールド hp, mp を初期化する public なコンストラクタ Mage( int hp, int mp )
  - ・自分の mp の 1/5 だけ攻撃相手(仮引数 c )の defence メソッドに渡すようにオーバーライドした public な attack メソッド。
  - ・受けたダメージ量(仮引数 damage )から自分の mp の 1/5 だけ引いた値が0以上の場合には、その値の分だけ自分の hp を減らし、さらに自分の mp を1だけ減らすようにオーバーライドした public な defence メソッド。
  - ・魔法使いを表す文字列 "魔法使い" を System.out.println メソッドで表示するようにオーバーライドした public な draw メソッド
- (e)動作テストを行う public なクラス Test。このクラスはファイル Test.java で定義され、その内容は解答用紙2枚目の冒頭のようになっている。

- (1) 以上のクラス及びインタフェイス(a)~(d)が書かれた各ファイルの内容を、解答欄(1)-(a)~(d)に書け。必要に応じて、package 宣言や import 宣言を記述するのを忘れないように。また、あるクラスの private メンバの内容を他のクラスから利用する場合には、用意されたアクセスを使うことも忘れないこと。なお、(e)の Test クラスが定義されている Test.java の内容は解答用紙2枚目に記載されているので参考にせよ。
- (2) RPGCharacter, Drawable, Knight, Mage の継承関係・実装関係を表す図を解答欄(2)に書け。なお、継承関係・実装関係を図で表す場合は矢印を使う。解答欄のクラス名・インタフェイス名の間を適切な矢印で結ぶこと。
- (3) Test クラスの main メソッドで、letItFight( ) メソッドが Knight 型オブジェクトと Mage 型オブジェクトを受け取れるのは何故か。また、letItDraw( ) メソッドが Knight 型オブジェクトと Mage 型オブジェクトを受け取れるのは何故か。解答欄(3)に論述せよ。

■問題2(5点)

どのような部分(変数やメソッドなど)がモジュールの実装部分になり、どのような部分(変数やメソッドなど)がモジュールのインタフェイス部分になるか、その考え方を答えよ。そして、問題1の RPGCharacter クラスの場合はどのメンバが実装部分やインタフェイス部分に相当するか、具体的にメンバ名をあげて述べよ。解答欄(4)に論述せよ。

■問題 1 ※スペースが足りない場合は、「裏へ続く」と書いて裏面に続きを書け。

解答欄 (1)-(a) RPGCharacter.java の内容 30 点 (部分点有り)

---

解答欄 (1)-(d) Mage.java の内容 20 点 (部分点有り)

---

解答欄 (1)-(b) Drawable.java の内容 10 点 (部分点有り)

---

解答欄 (1)-(c) Knight.java の内容 20 点 (部分点有り)

参考 : Test.java の内容 (e)

```
import com.tuis.rpg.character.RPGCharacter;
import com.tuis.rpg.system.Drawable;

public class Test {

    static void letItFight( RPGCharacter c1,
                           RPGCharacter c2 ) {
        while( c1.getHp() > 0 && c2.getHp() > 0 ) {
            c1.attack( c2 ); c2.attack( c1 );
        }
    }
}
```

```
static void letItDraw( Drawable d ) {
    d.draw();
}

public static void main( String args[] ) {
    Knight k = new Knight( 50, 10 );
    Mage m = new Mage( 30, 40 );
    letItDraw( k ); letItDraw( m );
    System.out.println( "現在のキャラクターの総数は"
        + RPGCharacter.getNum() );
    letItFight( k, m );
}
}
```

---

■問題 1 解答欄 (2) 10 点 (部分点有り)

---

■問題 1 解答欄 (3) 5 点 (部分点有り)

---

■問題 2 解答欄 (4) 5 点 (部分点有り)