

## 解答上の注意

- 1 解答は、解答用紙の問題番号に対応した解答欄にマークしなさい。
- 2 選択肢は大問ごとに用意されています。大問1の選択肢は大問2で使用しません。
- 3 選択肢は量が多いため、探しやすさの観点よりグループ分けされています。グループ分けに合わせて解答欄が区切られていますが、横1列で問題1つ分となっています。
- 4 問題の文中の【1】、【2】などには、特に指示がないかぎり、与えられた大問選択肢群が入ります。
- 5 出力結果選択肢群から選択する際は、基本データ型等の欄に答えなさい。
- 6 プログラムの穴埋めは授業の方法に準拠したものを選択肢から答えなさい。
- 7 同一の問題文中に【1】、【2】などが二度以上現れる場合、原則として、二度目以降は、同じ解答がはいります。

例題【1】に基本データ型等の③と答えたいとき

```
public static 【1】 main(String args[]){
    System.out.println("Hello World");
}
```

## マークシート例

基本データ型等	クラス名	変数	メソッド	ArrayList
[1] ① ② ③ ④ ⑤ ⑥	[1] ① ② ③ ④ ⑤ ⑥	[1] ① ② ③ ④ ⑤ ⑥	[1] ① ② ③ ④ ⑤ ⑥	[1] ① ② ③ ④ ⑤ ⑥
[2] ① ② ③ ④ ⑤ ⑥	[2] ① ② ③ ④ ⑤ ⑥	[2] ① ② ③ ④ ⑤ ⑥	[2] ① ② ③ ④ ⑤ ⑥	[2] ① ② ③ ④ ⑤ ⑥
[3] ① ② ③ ④ ⑤ ⑥	[3] ① ② ③ ④ ⑤ ⑥	[3] ① ② ③ ④ ⑤ ⑥	[3] ① ② ③ ④ ⑤ ⑥	[3] ① ② ③ ④ ⑤ ⑥

【大問選択肢群】

[基本データ型等]

- ① int
- ② String
- ③ void

[クラス名]

- ① Teacher
- ② Student

[変数]

- ① name
- ② id

[メソッド]

- ① getName
- ② getId

[ArrayList]

- ① get
- ② size

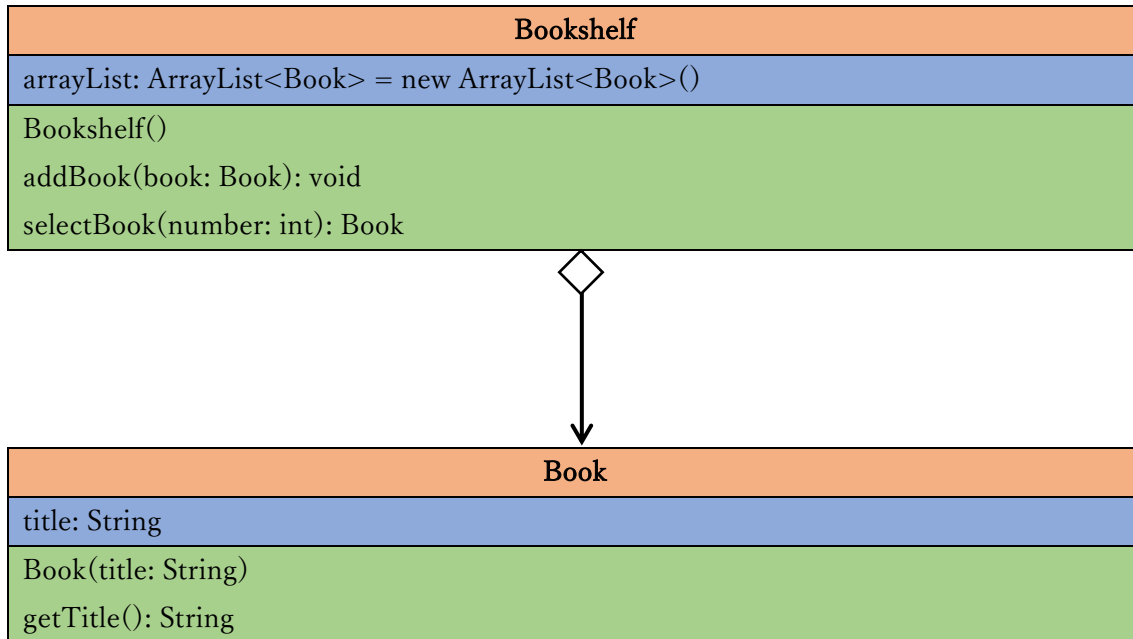
### 【大問1】

以下のクラス図と API 仕様から Bookshelf クラス、Book クラスの実装を行った。以下の

【1】～【9】に当てはまるものを選択肢から選びなさい。

※【1】～【8】大問選択肢群、【9】は出力結果選択肢群から選びなさい。

### クラス図



API 仕様 Bookshelf	
Bookshelf	コンストラクタです。
addBook	リストに Book オブジェクトを追加します。
selectBook	リストから指定された位置にある Book オブジェクトを取り出します。取り出された Book オブジェクトはリスト内から無くなります。

API 仕様 Book	
Book	title を引数で受け取るコンストラクタです。
getTitle	状態 title を返却します。

ArrayList の API の一部	
ArrayList	コンストラクタの一つです。
add	リストの最後に指定された要素を追加します。
get	リスト内の指定された位置にある要素を返します。ただし順番は 0 から始まります。
size	リスト内にある要素の数を返します。
remove	リストの指定された位置にある要素を削除し、要素を返却します。

#### Bookshelf のソースコード

```
import java.util.ArrayList;

public class Bookshelf{
    private ArrayList<Book> arrayList = new ArrayList<Book>();
    public Bookshelf(){
    }
    public void addBook(Book book){
        this.arrayList. 【1】 (book);
    }
    public Book selectBook(int number){
        return this.arrayList. 【2】 (number);
    }
}
```

#### Book のソースコード

```
public class Book{
    private String title;
    public Book(String title){
        this. 【3】 = 【3】 ;
    }
    public String getTitle(){
        return this. 【4】 ;
    }
}
```

A\_1 のソースコード

```
public class A_1{
    public static void main(String[] args){
        【5】 b = new 【5】 ();
        b.addBook(new 【6】 ("人間失格"));
        b.addBook(new 【6】 ("坊ちゃん"));
        System.out.println(b.【7】 .【8】 );
    }
}
```

A\_1 出力結果例

```
> java A_1
坊ちゃん
```

以下のコードを実行した際の出力結果として正しいものを【9】に答えなさい。

※この問題は基本データ型等にマークしなさい。

A\_2 のソースコード

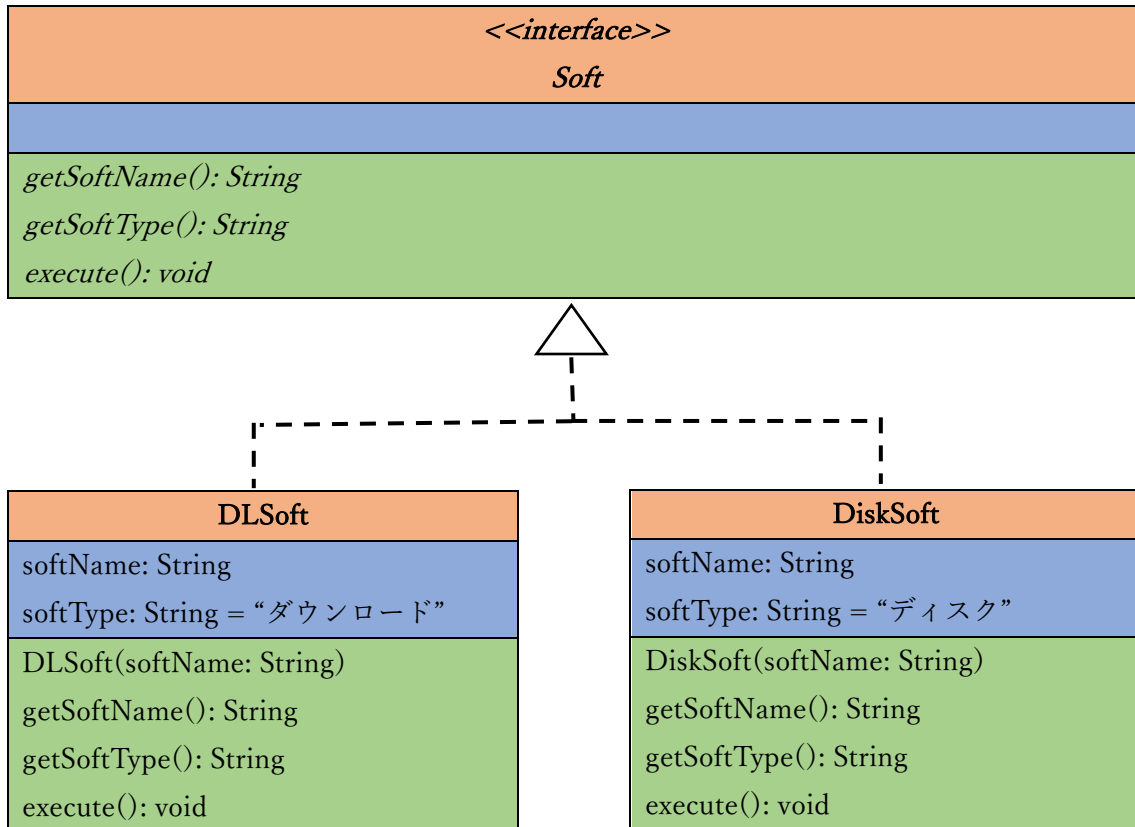
```
public class A_2{
    public static void main(String[] args){
        Bookshelf bookshelf = new Bookshelf();
        bookshelf.addBook(new Book("人間失格"));
        bookshelf.addBook(new Book("坊ちゃん"));
        System.out.println(s.selectBook(1).getTitle());
        bookshelf.addBook(new Book("こころ"));
        bookshelf.addBook(new Book("銀河鉄道の夜"));
        System.out.println(s.selectBook(2).getTitle());
    }
}
```

【9】 出力結果選択肢群

①	人間失格 坊ちゃん	②	人間失格 こころ
③	坊ちゃん こころ	④	坊ちゃん 銀河鉄道の夜

【大問2】

以下のクラス図と API 仕様から Soft クラス、DLSoft クラス、DiskSoft クラス、PS4 クラスの実装を行った。以下の【10】～【17】に当てはまるものを選択肢から選びなさい。  
 ※【10】～【12】，【14】～【16】は大問選択肢群、【13】，【17】はそれぞれの出力結果選択肢群から選びなさい。



API仕様 DLSoft	
DLSoft	softName を引数で受け取るコンストラクタです。
getSoftName	状態 softName を返却します。
getSoftType	状態 softType を返却します。
execute	ソフトを実行します。“データを読み込みます”と表示します。

API仕様 DiskSoft	
DiskSoft	softName を引数で受け取るコンストラクタです。
getSoftName	状態 softName を返却します。
getSoftType	状態 softType を返却します。
execute	ソフトを実行します。“ディスクを読み込みます”と表示します。

## Soft のソースコード

```
public interface Soft {  
    public String getSoftName();  
    public String getSoftType();  
    public void execute();  
}
```

## DLSoft のソースコード

```
public class DLSoft implements Soft{  
    private String softName;  
    private String softType = "ダウンロード";  
    public DLSoft(String softName){  
        this.【1 0】 = 【1 0】 ;  
    }  
    public String getSoftName(){  
        return this.【1 1】 ;  
    }  
    public String getSoftType(){  
        return this.【1 2】 ;  
    }  
    public void execute(){  
        System.out.println("データを読み込みます");  
    }  
}
```

DiskSoft のソースコード

```
public class DiskSoft implements Soft{
    private String softName;
    private String softType = "ディスク";
    public DiskSoft(String softName){
        this.【1 0】 = 【1 0】 ;
    }
    public String getSoftName(){
        return this.【1 1】 ;
    }
    public String getSoftType(){
        return this.【1 2】 ;
    }
    public void execute(){
        System.out.println("ディスクを読み込みます");
    }
}
```

**【出力結果】**

以下のコードを実行した際の出力結果として正しいものを【1 3】に答えなさい。

※この問題は基本データ型等にマークしなさい。

B\_1 のソースコード

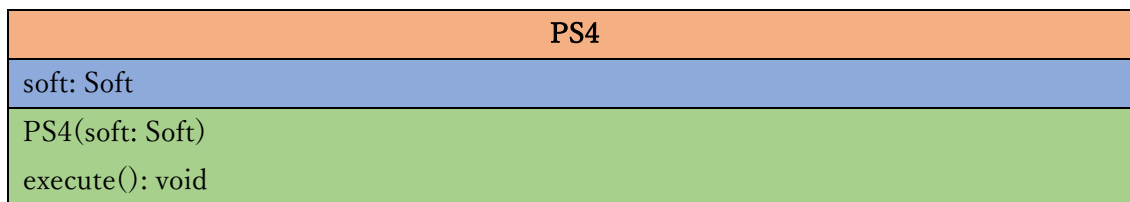
```
public class B_1{
    public static void main(String[] args){
        Soft soft1 = new DiskSoft("ドラクエ");
        Soft soft2 = new DLSoft("FF");
        System.out.println(soft1.getSoftType() + " " + soft1.getSoftName());
        System.out.println(soft2.getSoftType() + " " + soft2.getSoftName());
    }
}
```

**【1 3】 出力結果選択肢群**

①	ディスク ドラクエ ダウンロード FF	②	ダウンロード ドラクエ ディスク FF
③	ディスク FF ダウンロード ドラクエ	④	ダウンロード FF ディスク ドラクエ



## クラス図



API仕様 PS4	
PS4()	Soft オブジェクトを引数で受け取るコンストラクタです。
execute()	softName とソフトの実行時の処理を表示します。 例：FF のディスクを読み込みます

## PS4 のソースコード

```
public class PS4{
    private Soft soft;
    public PS4(Soft soft){
        this.【1 4】 = 【1 4】 ;
    }
    public void execute(){
        System.out.print(this.soft.【1 5】 + "の");
        this.soft.【1 6】 ;
    }
}
```

以下のコードを実行した際の実出力結果として正しいものを【17】に答えなさい。

※この問題は基本データ型等にマークしなさい。

B\_2 のソースコード

```
public class B_2{
    public static void main(String[] args){
        PS4 ps4 = new PS4(new DLSoft("ドラクエ"));
        ps4.execute();
        ps4 = new PS4(new DiskSoft("FF"));
        ps4.execute();
    }
}
```

【17】 出力結果選択肢群

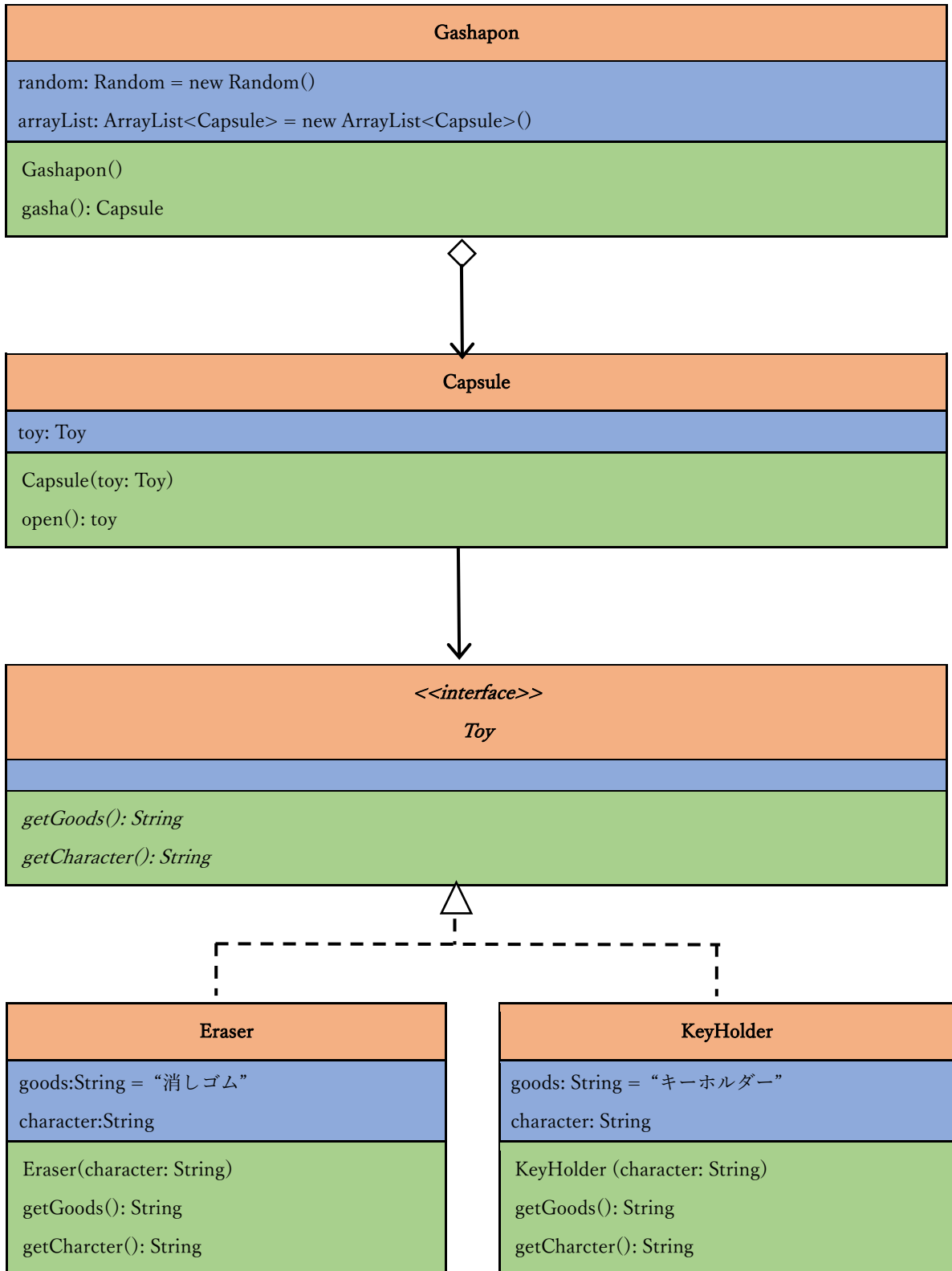
①	FF のデータを読み込みます ドラクエのディスクを読み込みます	②	ドラクエのデータを読み込みます FF のディスクを読み込みます
③	ドラクエのディスクを読み込みます FF のデータを読み込みます	④	FF のディスクを読み込みます ドラクエのデータを読み込みます

**【大問3】**

以下のクラス図と API仕様から Eraser クラス、KeyHolder クラス、Capsule クラス、Gashapon クラスの実装を行った。以下の【18】～【34】に当てはまるものを選択肢から選びなさい。

※【18】～【31】は大問選択肢群、【32】～【34】はそれぞれの出力結果選択肢群から選びなさい。

クラス図



## API 仕様

API 仕様 Eraser	
Eraser	character を引数で受け取るコンストラクタです。
getGoods	状態 goods の値を返却します。
getCharacter	状態 character の値を返却します。

API 仕様 KeyHolder	
KeyHolder	character を引数で受け取るコンストラクタです。
getGoods	状態 goods の値を返却します。
getCharacter	状態 character の値を返却します。

API 仕様 Capsule	
Capsule	Toy インターフェースを実装したクラスのオブジェクトを引数で受け取るコンストラクタです。
open	カプセルの中身を返却します。

API 仕様 Gashapon	
Gashapon	コンストラクタです。 名前がキン肉マンの消しゴム、名前がドラゴンボールの消しゴム、名前がハローキティのキーホルダーの3個の Toy インターフェースを実装したオブジェクトをカプセルに入れて ArrayList オブジェクトに格納します。
gasha	ArrayList オブジェクトからカプセルを一つ取り出し、返却します。取り出された Capsule オブジェクトはリスト内から無くなります。取り出すカプセルは疑似乱数で決定します。

ArrayList の API の一部	
ArrayList	コンストラクタの一つです。
add	リストの最後に指定された要素を追加します。
get	リスト内の指定された位置にある要素を返します。ただし順番は 0 から始まります。
size	リスト内にある要素の数を返します。
remove	リストの指定された位置にある要素を削除し、要素を返却します。

Random の API の一部	
Random	コンストラクタの一つです。
nextInt	疑似乱数を生成します。ただし、0 以上で引数より小さい整数です。

### Toy のソースコード

```
public interface Toy{
    public String getGoods();
    public String getCharacter();
}
```

### Eraser のソースコード

```
public class Eraser 【 1 8 】 Toy{
    private String goods = "消しゴム";
    private String character;
    public Eraser(String character){
        this. 【 1 9 】 = 【 1 9 】 ;
    }
    public String getGoods(){
        return this. 【 2 0 】 ;
    }
    public String getCharacter(){
        return this. 【 2 1 】 ;
    }
}
```

### KeyHolder のソースコード

```
public class KeyHolder 【 1 8 】 Toy{
    private String goods = "キーホルダー";
    private String character;
    public KeyHolder(String character){
        this. 【 1 9 】 = 【 1 9 】 ;
    }
    public String getGoods(){
        return this. 【 2 0 】 ;
    }
    public String getCharacter(){
        return this. 【 2 1 】 ;
    }
}
```

### Capsule のソースコード

```
public class Capsule{
    private Toy toy;
    public Capsule(Toy toy){
        this.【2 2】 = 【2 2】 ;
    }
    public Toy open(){
        return this.【2 3】 ;
    }
}
```

### Gashapon のソースコード

```
import java.util.ArrayList;
import java.util.Random;

public class Gashapon{
    private Random random = new Random();
    private ArrayList<Capsule> arrayList = new ArrayList<Capsule>();
    public Gashapon(){
        this.arrayList.add(new Capsule(new 【2 4】 ("キン肉マン")));
        this.arrayList.add(new Capsule(new 【2 5】 ("ドラゴンボール")));
        this.arrayList.add(new Capsule(new 【2 6】 ("ハローキティ")));
    }
    public Capsule gasha(){
        【2 7】 a = this.random.nextInt(this.arrayList.size());
        【2 8】 b = this.arrayList.【2 9】 (a);
        return b;
    }
}
```



#### C\_1 のソースコード

```
public class C_1 {  
    public static void main(String[] args) {  
        Gashapon gashapon = new Gashapon();  
        Capsule capsule = gashapon. 【 3 0 】 ();  
        Toy toy = 【 3 1 】 .open();  
        System.out.print("カプセルの中身は");  
        System.out.print(toy.getCharacter());  
        System.out.println(toy.getGoods() + "です。");  
    }  
}
```

#### C\_1 出力結果例

```
>java C_1  
カプセルの中身はドラゴンボール消しゴムです。  
  
>java C_1  
カプセルの中身はハローキティキーホルダーです。  
  
>java C_1  
カプセルの中身はキン肉マン消しゴムです。
```

**【出力結果】**

以下のコードを実行した際の出力結果として正しいものを【32】に答えなさい。

※この問題は基本データ型等にマークしなさい。

C\_2のソースコード

```
public class C_2{
    public static void main(String[] args){
        Toy toy1 = new Eraser("キン肉マン");
        Toy toy2 = new KeyHolder("ハローキティ");
        System.out.println(toy1.getGoods() + " " + toy1.getCharacter());
        System.out.println(toy2.getGoods() + " " + toy2.getCharacter());
    }
}
```

**【32】出力結果選択肢群**

①	消しゴム キン肉マン 消しゴム ハローキティ	②	キーホルダー キン肉マン キーホルダー ハローキティ
③	消しゴム キン肉マン キーホルダー ハローキティ	④	キーホルダー キン肉マン 消しゴム ハローキティ

以下のコードを実行した際の出力結果として正しいものを【33】に答えなさい。

※この問題は基本データ型等にマークしなさい。

C\_3のソースコード

```
public class C_3{
    public static void main(String[] args){
        Toy toy1 = new Eraser("キン肉マン");
        Capsule capsule = new Capsule(toy1);
        Toy toy2 = capsule.open();
        System.out.println("カプセルの中身は" + toy2.getCharacter() +
                            toy2.getGoods() + "です。");
    }
}
```

【33】出力結果選択肢群

①	カプセルの中身はキン肉マン消しゴムです。	②	カプセルの中身はキン肉マンキーホルダーです。
③	カプセルの中身は消しゴムです。	④	カプセルの中身はキン肉マンです。

以下のコードを実行した際の出力結果として正しくないものを【34】に答えなさい。

※この問題は基本データ型等にマークしなさい。

C\_4のソースコード

```
public class C_4{
    public static void main(String[] args){
        Gashapon gashapon = new Gashapon();
        Capsule capsule = gashapon.gasha();
        Toy toy = capsule.open();
        System.out.println("カプセルの中身は" + toy.getCharacter() + toy.getGoods()
                            + "です。");
    }
}
```

【34】出力結果選択肢群

①	カプセルの中身はハローキティキーホルダーです。	②	カプセルの中身はキン肉マン消しゴムです。
③	カプセルの中身はドラゴンボール消しゴムです。	④	カプセルの中身はドラゴンボールキーホルダーです。

<p><b>【大問1 選択肢群】</b></p> <p><b>[基本データ型等]</b></p> <ul style="list-style-type: none"> <li>① int</li> <li>② String</li> <li>③ void</li> <li>④ implements</li> <li>⑤ extends</li> <li>⑥ this</li> </ul> <p><b>[クラス名]</b></p> <ul style="list-style-type: none"> <li>① Book</li> <li>② Bookshelf</li> </ul> <p><b>[変数]</b></p> <ul style="list-style-type: none"> <li>① bookshelf</li> <li>② book</li> <li>③ title</li> <li>④ number</li> </ul> <p><b>[メソッド]</b></p> <ul style="list-style-type: none"> <li>① selectBook()</li> <li>② selectBook(1)</li> <li>③ selectBook(2)</li> <li>④ getTitle()</li> <li>⑤ addBook()</li> <li>⑥ addBook(1)</li> </ul> <p><b>[ArrayList]</b></p> <ul style="list-style-type: none"> <li>① get</li> <li>② size</li> <li>③ remove</li> <li>④ add</li> </ul>	<p><b>【大問2 選択肢群】</b></p> <p><b>[基本データ型等]</b></p> <ul style="list-style-type: none"> <li>① int</li> <li>② String</li> <li>③ void</li> <li>④ implements</li> <li>⑤ extends</li> <li>⑥ this</li> </ul> <p><b>[クラス名]</b></p> <ul style="list-style-type: none"> <li>① Soft</li> <li>② DLSoft</li> <li>③ DiskSoft</li> <li>④ PS4</li> </ul> <p><b>[変数]</b></p> <ul style="list-style-type: none"> <li>① softName</li> <li>② softType</li> <li>③ soft</li> </ul> <p><b>[メソッド]</b></p> <ul style="list-style-type: none"> <li>① getSoftName()</li> <li>② getSoftTye()</li> <li>③ execute()</li> </ul> <p><b>[ArrayList]</b></p> <ul style="list-style-type: none"> <li>① get</li> <li>② size</li> <li>③ remove</li> <li>④ add</li> </ul>
--	---

**【大問3 選択肢群】**

**[基本データ型等]**

- ① int
- ② String
- ③ void
- ④ implements
- ⑤ extends
- ⑥ this

**[クラス名]**

- ① Toy
- ② Eraser
- ③ KeyHolder
- ④ Capsule
- ⑤ Gashapon

**[変数]**

- ① goods
- ② chaacter
- ③ toy
- ④ capsule

**[メソッド]**

- ① getGoods
- ② getCharacter
- ③ open
- ④ gasha

**[ArrayList]**

- ① get
- ② size
- ③ remove
- ④ add