

1. 次のような記述がある。

```
class Person {
    int name;
}
class Member {
    Person leader = new Person();
    Person subleader = new Person();
}
```

この場合、「Person」は (7) といい、「leader」は「Person」を具体化したもので (8) という。

2. 次のような記述がある。

```
class Vehicle {
    int speed;
    void setSpeed( int s ) { speed = s; }
}
```

この場合、「speed」は (9) で、「setSpeed」は (10) である。データと機能を一つにまとめて、オブジェクトの内部を見えなくすることを (11) といい、これにより外部からのアクセスを制限することができる。これをデータ隠蔽という。

3. 異なるオブジェクトに同じメッセージを送ると、それぞれのオブジェクトに応じた処理が行われる性質を (12) (多態性) という。

4. クラス `Animal` を基に、新しい性質を付加したクラス `Dog` を定義することを、 という。このとき、 は のサブクラスである。Java では、あるクラスのサブクラスは、 定義することができる。

(7) ~ (10) の解答群

(15) の解答群

ア インスタンス	イ クラス	ア クラス Animal	イ クラス Animal のサブクラス
ウ メソッド	エ メンバ変数	ウ クラス Dog	エ クラス String

(11) ~ (13) の解答群

(16) の解答群

ア インスタンス化	イ カプセル化	ア 一つだけ	イ 二つまで
ウ ポリモフィズム	エ 継承	ウ 三つまで	エ いくつでも

(14) の解答群

ア クラス Animal イ クラス Dog のスーパークラス
ウ クラス Dog エ クラス Object

(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
-----	-----	-----	------	------	------	------	------	------	------

- 問2 次の文章はオブジェクト指向プログラミングについて説明したものである。□に入る適切な字句を、解答群の中から選べ。

1. 次のような記述がある。

```
class Square {
    int sideLength;
    int getSideLength() { return sideLength; }
}

class Pattern {
    Square part = new Square();
}
```

この場合、「Square」は (7) で、生成した (8) を格納している「part」は参照型の変数である。このように、(8) を生成することを (9) という。

2. 次のような記述がある。

```
class Counter {
    int countNum;
    void countUp() { countNum++; }
}
```

このとき、countNumを (10)，countUpを (11) という。

3. オブジェクトには、「自律的に振舞う」という特徴がある。これは、オブジェクト A がオブジェクト B にメッセージを送る場合、その処理の内容を知っているのは (12) ということである。

4. すでに定義されているクラスを拡張して、新しいクラスを定義することを (13) という。このとき、既存のクラスを (14) といい、拡張して新しく定義したクラスを (15) という。

5. 異なる種類のオブジェクトに同じメッセージを送ると、それぞれのオブジェクトに応じた処理が行われる性質を (16) (多態性) という。

(7), (8) の解答群

(12) の解答群

(16) の解答群

ア	クラス	イ	メソッド	ア	オブジェクト A	ア	ポリモフィズム
ウ	インスタンス	エ	モジュール	イ	オブジェクト B	イ	カプセル化

- (9) の解答群

ウ	オブジェクトAを作成したオブジェクト	ウ	抽象化
エ	オブジェクトBを作成したオブジェクト	エ	継承

ア カプセル化 イ インスタンス化
ウ モジュール化 エ 現実化

(13) の解答群

- (10), (11) の解答群

ア 継承 イ カプセル化
ウ コンポーネント エ インプリメント

ア 属性 イ 実装
ウ 関係 エ 操作

(14), (15) の解答群

ア 類似クラス	イ サブクラス
ウ 原始クラス	エ スーパークラス

(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
-----	-----	-----	------	------	------	------	------	------	------