

問1 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 Satellite {
    long mass;
}
class Mars {
    Satellite phobos = new Satellite();
    Satellite deimos = new Satellite();
}
```

この場合、Satellite は (7) といい、phobos は Satellite を実体化した (8) を参照する変数である。

2. 次のような記述がある。

```
class Counter {
    private int count;
    void reset() { count = 0; }
    void countup() { count++; }
    int getCount() { return count; }
}
```

この場合、count は (9) で、getCount 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。これを情報隠ぺいという。

3. 共通した性質を親クラスにまとめることを (12) という。また、独自の性質をもった subclasses を作成することを (13) という。

4. すでに定義されているクラスの機能や属性を引き継いで、新しいクラスを定義することができる。これを (14) という。このとき、既存のクラスを (15) といい、新しく定義したクラスを (16) という。

(12), (13) の解答群

ア クラス	イ メソッド	ア 特化	イ 汎化
ウ インスタンス	エ メッセージ	ウ 抽象化	エ インスタンス化

(14) の解答群

ア	インスタンス	イ	メソッド	ア	継承	イ	ポリモフィズム
ウ	クラス	エ	メンバ変数	ウ	カプセル化	エ	インスタンス

(15), (16) の解答群

ア	ポリモフィズム	イ	カプセル化	ア	サブクラス	イ	カスタムクラス
ウ	インスタンス化	エ	継承	ウ	スーパークラス	エ	ベースクラス

(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 Client {
    void link() {
        Connection con = new Connection();
    }
}
```

この場合、(7) はクラス名で、(8) は参照型の変数名である。

2. 次のような記述がある。

```
class Record {
    int id;

    void setID(int id) {
        this.id = id;
    }
}
```

id を (9) , setID() を (10) という。このように、(9) へのアクセスをメソッド経由に限定し、外部から隠ぺいすることを (11) 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. すでに定義されているクラスをもとに機能や性質の追加や変更を行って、新しいクラスを定義することができる。これを (13) という。Watch クラスをもとに AnalogWatch クラスを定義した場合、Watch クラスを (14) , AnalogWatch クラスを (15) という。このとき、既存のクラスに新しい性質や機能を加えることを (16) という。

(13) の解答群

ア Connection	イ con	ア 汎化	イ 継承
ウ link	エ void	ウ クラス化	エ インスタンス化

(14), (15) の解答群

ア サブクラス	イ メンバ変数	ア メソッド	イ サブクラス
ウ スーパークラス	エ メソッド	ウ オブジェクト	エ スーパークラス

(16) の解答群

ア 汎化	イ カプセル化	ア 汎化	イ カプセル化
ウ 継承	エ 共通化	ウ 拡張	エ インスタンス化

(12) の解答群

ア ポリモフィズム      イ 継承  
ウ インスタンス化      エ カプセル化

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