Exercícios

Tente resolver manualmente e, em seguida, programe o código para descobrir as respostas

Exercício: desenhe *diagramas de classe* para cada um dos conjuntos de classes e interfaces conforme o exemplo e legenda abaixo.

1.
```
public interface Foo { }
public class Bar implements Foo { }
```

2.
```
public interface Vinn { }
public abstract class Vout implements Vinn { }
```

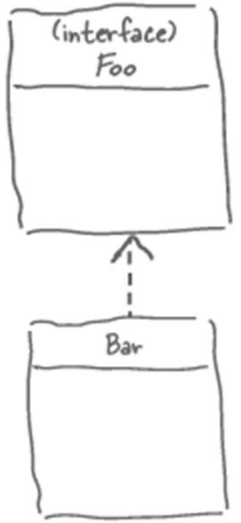
3.
```
public abstract class Muffie implements Whuffie { }
public class Fluffie extends Muffie { }
public interface Whuffie { }
```

4.
```
public class Zoop { }
public class Boop extends Zoop { }
public class Goop extends Boop { }
```

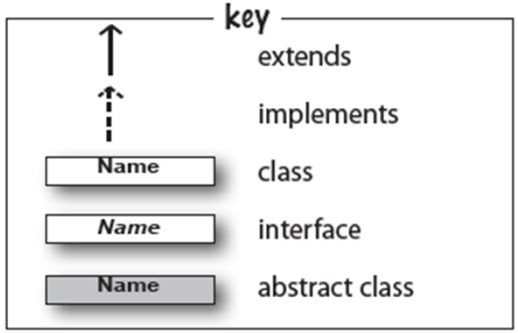
5.
```
public class Gamma extends Delta implements Epsilon { }
public interface Epsilon { }
public interface Beta { }
public class Alpha extends Gamma implements Beta { }
public class Delta { }
```

Exemplo:

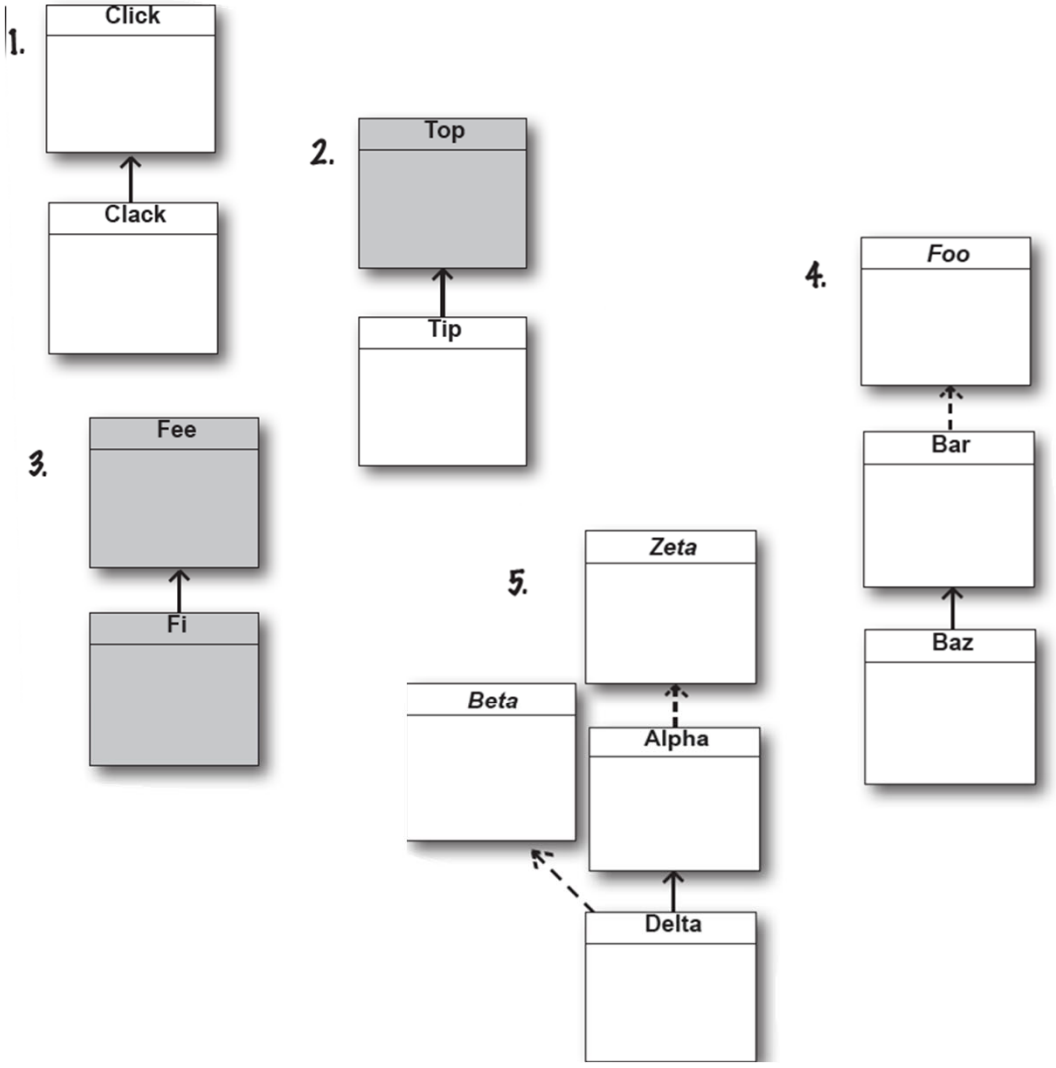
1.



Legenda:



Exercício: Dado os *diagramas de classe* abaixo, determine os conjuntos de classes e interfaces conforme o exemplo e legenda abaixo.

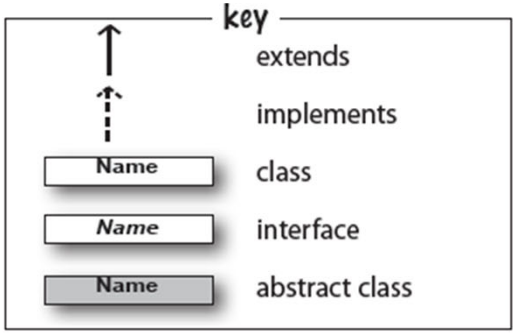


Exemplo:

1. `public class Click { }`

   `public class Clack extends Click { }`

Legenda:



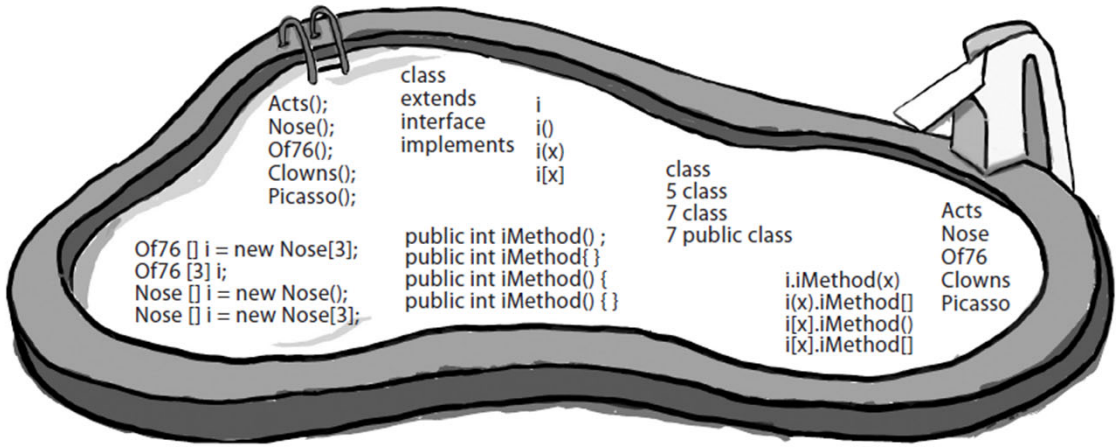
Exercício: determine aplicações em que as classes abaixo podem ser concretas e abstratas conforme os exemplos.

Concrete	Sample class	Abstract
golf course simulation	Tree	tree nursery application
_____	House	architect application
satellite photo application	Town	_____
_____	Football Player	coaching application
_____	Chair	_____
_____	Customer	_____
_____	Sales Order	_____
_____	Book	_____
_____	Store	_____
_____	Supplier	_____
_____	Golf Club	_____
_____	Carburetor	_____
_____	Oven	_____

Exercício: pegue trechos de código da piscina e coloque-os nas linhas em branco do código de modo que o conjunto de classes seja compilado, executado e produza o resultado abaixo.

```
_____ Nose {
  _____
}

abstract class Picasso implements _____{
  _____
    return 7;
  }
}

class _____ _____ _____ { }

class _____ _____ _____ {
  _____
    return 5;
  }
}
```

```
public _____ _____ extends Clowns {

  public static void main(String[] args) {
    _____
    i[0] = new _____
    i[1] = new _____
    i[2] = new _____
    for (int x = 0; x < 3; x++) {
      System.out.println(_____
            + " " + _____.getClass());
    }
  }
}
```



Um trecho de código pode ser usado mais de uma vez, e nem todos são necessários.

Resultado

```
File  Edit  Window  Help  BeAfraid
%java _____
5 class Acts
7 class Clowns
        Of76
_____
```