## NAME

Lintian::DepMap::Properties – Dependencies with properties map/tree creator

## SYNOPSIS

```
use Lintian::DepMap::Properties;

my $map = Lintian::DepMap::Properties->new;
```

## DESCRIPTION

Lintian::DepMap::Properties is a simple layer between Lintian::DepMap and the application allowing nodes to have application-defined properties.

add(node, [dependencies], [ref to property])

Adds a node with possibly one or more dependencies and sets the `node`'s property to the ref, if defined. The property must be a reference (it can be to a hash, an array, a function, an object, etc) and must be the last argument given to the method.

E.g.

```
$map->add('foo', {name => 'John Doe', age => 20});
```

addp(node[, prefix, dependency[, dependency...]], [ref to property])

Adds the given `node` to the map marking any third or more parameters, after prefixing them with `prefix`, as its dependencies and sets the `node`'s property to the ref, if defined. See **add()**'s description for more information about properties. E.g.

```
# pA and pB have no dependency:
$map->addp('pA', {name => 'John Doe'});
$map->addp('pB', {name => 'Jane Doe'});
# Df depends on pA and pB:
$map->addp('Df', 'p', 'A', 'B', {name => 'Doe Family'});
```

getp(node)

Returns the reference to the given `node`'s properties.

E.g.

```
# prints John Doe
print $map->getp('foo')->{'name'};
# changes the value of 'name'
$map->getp('foo')->{'name'} = 'Jane Doe';
# prints Jane Doe
print $map->getp('foo')->{'name'};
```

## AUTHOR

Originally written by Raphael Geissert <atomo64@gmail.com> for Lintian.