

Dziedziczenie

Wszystkie klasy, na bazie których powstają obiekty, dziedziczą po klasie `Object`

```
1 public class Superbohater {  
2  
3     private int punktyZycia;  
4     private int sila;  
5     private int zrecznosc;  
6     private int wytrzymalosc;  
7  
8     public Superbohater() {}  
9  
10    public Superbohater(int punktyZycia, int sila, int zrecznosc, int wytrzymalosc) {  
11        this.punktyZycia = punktyZycia;  
12        this.sila = sila;  
13        this.zrecznosc = zrecznosc;  
14        this.wytrzymalosc = wytrzymalosc;  
15    }  
16  
17    // Gettery i settery...  
18 }
```




```

1  public class Batman extends Superbohater {
2
3      private boolean batmobil;
4
5      public Batman(int punktyZycia, int sila, int zrecznosc, int wytrzymalosc, boolean batmobil) {
6          super(punktyZycia, sila, zrecznosc, wytrzymalosc);
7          this.batmobil = batmobil;
8      }
9
10     public void uzyjBatmobilu() {
11         if (batmobil) {
12             System.out.println("Batmobil zostal uzyty.");
13         } else {
14             System.out.println("Batmobil zostal w garazu.");
15         }
16     }
17 }

```

super() wywołuje konstruktor klasy dziedziczonej

```

1  Batman batman = new Batman(300, 20, 15, 30, true);
2
3  // Metoda getPunktyZycia() jest zdefiniowana
4  // w klasie Superbohater.
5  int punktyZycia = batman.getPunktyZycia();

```



```
1 public class Spiderman extends Superbohater {  
2  
3     public Spiderman(int punktyZycia, int sila, int zrecznosc, int wytrzymalosc) {  
4         super(punktyZycia, sila, zrecznosc, wytrzymalosc);  
5     }  
6  
7     public void chodzPoScianach() {  
8         System.out.println("Chodzi po scianach.");  
9     }  
10  
11     public void wystrzelPajeczyny() {  
12         System.out.println("Pajeczyna wystrzelona.");  
13     }  
14 }
```

