Rady

- invariant cyklu

```
int b = …;
for (…) {
  int a1 = 150 * b;      // toto vyřeší „chytrý" kompilátor
  int a2 = 150 * f(b);   // ale toto už ne, ačkoli též může být
                         // a2 invariantní
}
```

- kompozice (obalení třídy) na místo dědění

dědění „cizí" třídy přináší riziko pozdější změny rodičovské třídy
programátor „uživatel" zděděné třídy nemůže volat nepřekryté metody rodičovské třídy, čímž by mohl ohrozit pozdější změny implementace zděděné třídy

```
public class A {
    public void add(int x) { … /* kód metody */ }
    public void add(int []arr)
       { for (int x:arr) { … /* totožný kód */ } }
}

public class B extends A {
  void log(int a)  { … }
  public void add(x) { log(x); super.add(x); }
  public void add(int []arr) { for (int x:arr) { log(x); }
     super.add(arr);  }
}
```

v tomto případě raději:

```
public class B {
  private A a;
  void log(int a)  { … }
  public void add(x) { log(x); a.add(x); }
  public void add(int []arr) { for (int x:arr) { log(x); }
     a.add(arr);  }
}
```

problém by nastal, pokud by později byla třída A změněna tak, že by druhá z metod začala volat tu první na místo provádění totožného kódu