JAVA: null this super

Mindaugas Karpinskas
2017

Trumai

- Specialūs kintamieji
  - null
  - this
  - super

Specialūs kintamieji

Java turi tris iš anksto apibrėžtus kintamuosius: null, this ir super. Pirmieji du yra Object tipo. null žymi neegzistuojantį objektą, o this nurodo tą patį objekto egzempliorių. super nurodo į tiesioginę superklasę. Pasižiūrėkime pavyzdį.

null kintamasis (1)

- Esama aptare, kad bet koks klasę žymintis kintamasis turi būti inicijuojamas.
- Jei jis nėra inicijuotas, jo reikšmė lygi **null** specialiam kintamajam.
- **null** objektas neturi jokių kintamųjų ir metodų, todėl su juo negalime atlikti jokių manipuliacijų.
    - Gan dažnai pasitaikančios programavimo kalidos, kai bandoma panaudoti neinicijuotą objektą. Tuomet gausite standartinį pranešimą: NullPointerException. Pateiktas žemiau pavyzdys parašytas kiek nekorektiškai, nes ReplaceChars metodas naudojamas nepatikrinus ar jis nėra null:

```
public void someMethod(ReplaceChars a) {
        a.replaceNextChar('a','b');
}
```

null kintamasis (2)

null panaudojimo pavyzdys, kurio vykdymas sustotų ir būtų gautas pranešimas NullPointerException:

```
ReplaceChars B;
someMethod(B);
```

Kad programa nesustotų, teks tikrinti ar objektai nėra null. Jeigu klasės tipo kintamasis nėra null - galėsime juo naudoti, priešingu atveju - nesinaudosime, kad negauti NullPointerException klaidos pranešimą. Perrašykime pavyzdį naujai:

```
public void someMethod(String s) {
    if (s == null) {
        System.out.println("s is null!!!");
    }else {
        System.out.println(s.length());
}
```

null kintamasis (2)

null panaudojimo pavyzdys, kurio vykdymas sustotų ir būtų gautas pranešimas NullPointerException:

```
ReplaceChars B;
someMethod(B);
```

Kad programa nesustotų, teks tikrinti ar objektai nėra null. Jeigu klasės tipo kintamasis nėra null - galėsime juo naudoti, priešingu atveju - nesinaudosime, kad negauti NullPointerException klaidos pranešimą. Perrašykime pavyzdį naujai:

```
public void someMethod(ReplaceChars A) {
    if (A==null) {
        System.out.println("A is null!!!");
    }else {
        A.replaceNextChar('a','b');
}
```

Užuotis Null1

- Sukurti metodą kuris turėtu du parametrus:
  - masyvas - String tipo masyvą
  - tekstas - String tipo parametras
- Metodas turi patalpinti parametro **tekstas** reikšmę į pirmą laisvą masyvo elementą
- Metodas turi grąžinti teigiamą rezultatą, boolean tipo, tik tuo atveju jei jam pavyko patalpinti rezultatą
- Metodas turi turėti patikrinimus ar su paduotais parametrais galima atlikti priskyrimą

- Main metode išbandyti metodą

ArrayHelper

array != **null - > false**

value tusčias -> false

Nerandu laisvos vietos -> **false**

```
class ArrayHelper {
	public boolean append(String[] array, String value) {
		if (value == null || value.trim().equals("")) {
			return true;
		}
		if (array != null) {
			for (int i = 0; i < array.length; i++) {
				if (array[i] == null) {
					array[i] = value;
					return true;
				}
			}
		}
		return false;
	}
}
```

NullTest

```
public class NullTest {
	public static void main(String[] args) {
		ArrayHelper helper = new ArrayHelper();

		String[] m = new String[2];
		System.out.println("1 -"+helper.append(m, "Aš"));
		System.out.println("2 -"+helper.append(m, " "));
		System.out.println("3 -"+helper.append(m, "Mindaugas"));
		System.out.println("4 -"+helper.append(m, "Taip"));

		System.out.println(Arrays.toString(m));
	}
}
```

1 -true

2 -true

3 -true

4 -false

[Aš, Mindaugas]

Null2

Sukurti metodą kuris išspausdina perduoto String **tekstas** parametro reikšmę tik tuo atveju, jei ji nelygi null.

Null3

Sukurti metodą kuris išspausdina perduoto String tipo masyvo elementų reišmes. Spausdina tik egzistuojančias reikšmes.

this

this kintamasis

- Kartais jums reikės perduoti nuorodą einamo objekto į metodą ar konstruktrių. Tuo atveju ir naudojamas **this**.
- Tarkime Son ir Daughter konstruktoriai naudoja Mom kintamąjį. Šiuo atveju ir pravers **this** kintamasis.

```
class Mom extends Object {
    // declarations,definitions
}
class Son {
    Mom myMommy;
    public Son(Mom mommy) {
        this.myMommy = mommy;
    }

}

class Daughter {
    Mom myMommy;
    public Daughter(Mom mommy) {
        this.myMommy = mommy;
    }
}
```

this kintamasis

Kai Mom konstruoja savo Sons ir Daughters, jei reikės perduoti nuorodą į save pačią. Praktiškai tai gali atrodys taip

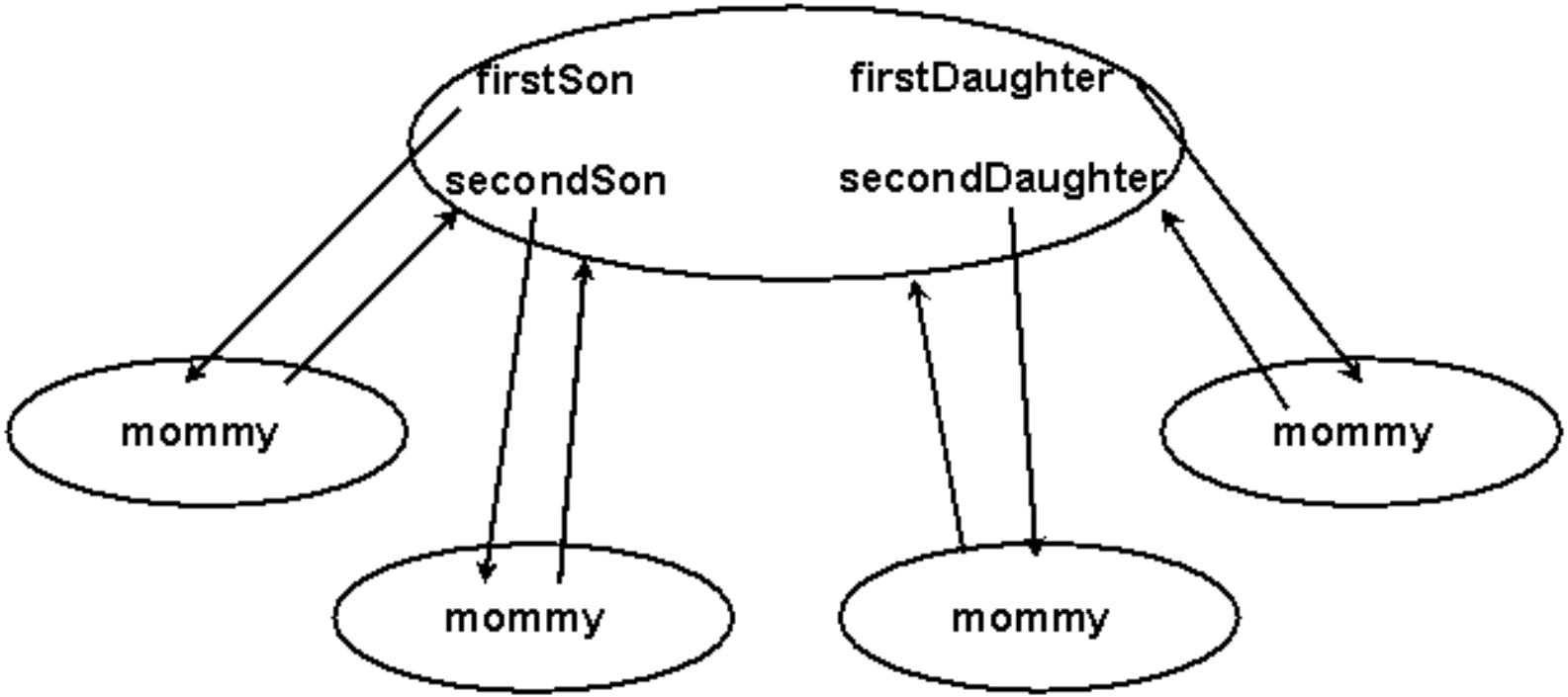
```
class Mom {
      Son firstSon;
      Son secondSon;
      Daughter firstDaughter;
      Daughter secondDaughter;

      public Mom() {
            firstSon = new Son(this);
            secondSon = new Son(this);
            firstDaughter = new Daughter(this);
            secondDaughter = new Daughter(this);
      }

      // other methods
}
```

Mom bigMama = new Mom();

Klasių hierarchiją



super

super kintamasis

super pagalba galime pasiekti tėvinės kalsės kintamosius ir metodus

```
class Mom {
        // declarations, constructors
        public void cleanUpRoom() {
                // code for cleaning up room
        }
        // other methods)
}
class Son extends Mom {
        // kintamieji, konstruktoriai
        public void cleanUpRoom() {
                super.cleanUpRoom();
                System.out.println("Cleaned up my room!!");
        }
        // other methods
}
```

Konstruktoriai, kaip ir metodai gali naudoti **super** kintamąjį

Mūsų poklasė gali panaudoti parašytą konstruktoriuje kodą panaudodama **super** kintamąjį.
Naudojant super kintamąjį su konstruktoriais reikia atsiminti porą išimčių. Pirma, šį kintamąjį galite naudoti tik konstruktoriaus kamiene. Antra, jis turi būti panaudotas pačioje konstruktoriaus kamieno pradžioje.

```
class SuperClass {
    private int onlyInt;
    public SuperClass(int i) {
        onlyInt = i;
    }
    public int getOnlyInt() {
        return onlyInt;
    }
}
class SubClass extends SuperClass {
    private int anotherInt;
    public SubClass(int i, int j) {
        super(i);
        anotherInt = j;
    }
    public int getAnotherInt() {
        return anotherInt;
    }
}
```

Uzd1

Dvi klasės: Automobilis, Audi;

Audi paveldi automobilį. Automobilis turi metodą: pirmyn(); - išvedą pranešima “Automobilis juda pirmyna”.

Audi perrašo metodą pirmyn(): - išveda “Audi lekia pirmyn!!!”

Audi turi metodą testas(); kuriame reikia iškviesti Audi metodą pirmyn() po to Automobilio metodą pirmyn();

Klausimai