PB162 Cvicenie 8. skupina 23

Polia, vynimky, metody equals() a hashCode()

Rastislav Mirek

Fakulta informatiky, Masarykova univerzita

9. novembra 2009

Polia

- Pole objektov(Person, String...) alebo pole zakladnych typov(int, double...)
- Pole nejakych typov(objektov alebo zakladnych) je samo o sebe typ. Moze sa pouzivat vsade kde ine typy(t.j. atribut, lokalna premenna a parametre mozu mat typ pole).
- Pole ako typ je objektovy typ.
- Instanciu pola vytvarame klucovym slovom new.
- Raz vytvorene pole(za pouzitia new) ma danu, nemennu dlzku (zadanu pri vytvoreni pola).
- priklady(syntax):
  - `private String[] allTexts = new String[15];`
    Pole najviac 15 retazcov ako atribut
  - `int[] computedNumbers = new int[computedCount];`
    Pole najviac `computedCount` integer-ov ako lokalna premenna
  - `Person[] myFriends;`
    Neinicializovane pole ludi ako lokalana premenna

Viacrozmerne polia

- Pole nejakych typov(objektov alebo zakladnych) je samo o sebe typ. Moze sa pouzivat vsade kde ine typy(t.j. atribut, lokalna premenna a parametre mozu mat typ pole).
- Pole ako typ je objektovy typ.
- $\Longrightarrow$moze byt pole objektov typu pole. T.j. pole poli, dvojrozmerne pole. Taketo pole je opat objektovym typom teda mozme mat aj trojrozmerne pole atd.
  - `private Car[][] parkplace;`
    Neinicializovane dvojrozmerne pole aut ako atribut.
  - `Car[][] parkplace = new Car[rows][columns];`
    Premenne rows columns su celociselneho typu. Dvojrozmerne pole aut ako lokalna premena. Prvy rozmer pola je `rows` (napr. pocet radov na parkovisku) druhy je `columns`(napr. pocet stlpcov na parkovisku).

Pristup k prvkom pola

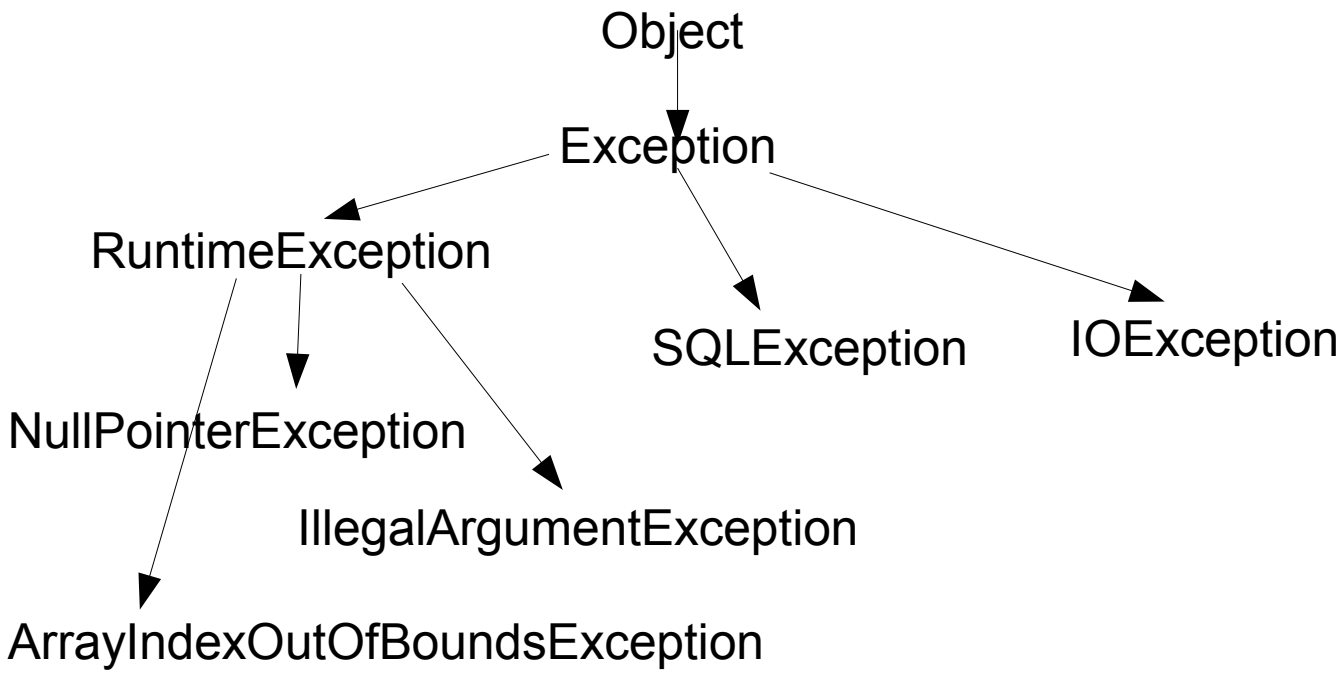
- Polia sa v Jave indexuju od nuly.
- po vytvoreni pola su vsetky prvky pola null.
- ```
  private String[] allTexts = new String[3];
  allTetxs[0] = “first text”;
  allTexts[1] = “second text”;
  allTexts[2] = “third text”;
  String choosenText = allTexts[1];
  ```
- ```
  int rows = 2; int columns = 2;
  Car[][] parkplace = new Car[rows][columns];
  parkplace[0][1] = new Car();
  ```
- S polami sa casto pracuje v cykloch. Dlzku pola nam vrati `allTexts.length;`
  - Nie je to metoda ale verejny atribut (ziadne zatvorky).
  - Neduplikujeme informaciu o dlzke pola tak ze si ju niekde ukladame
- Specialna syntax pre vycet prvkov pola pri jeho vytvoreni

Vynimky

- Osetruju chybove stavy. Chyby programatora, chyby pri komunikacii aplikacie (z OS, citanie suborov, inymi aplikaciami...).
- Naozaj len chybove stavy. Chybne vstupy od uzivatela by mali byt osetrene inak. Velmi narocne na vypocet.
- Vyhodenie vynimky:
  `throw new SomeKindOfException(<parameters>);`
- Vynimky runtime a ostatne. Vyhodenie ostatnych musime deklarovat v hlavicke metody. Klucove slovo `throws`
- Zachytenie vynimky:

```
try{
    //some code here can throw exception
} catch(SomeKindOfException ex){
    //exception ex handling code here
} finally{
    //this code is executed unconditionally
}
```

Hierarchia vynimiek



Prekryvanie equals

- Kazda trieda ma metodu `boolean equals(Object o)`. Dedi sa z triedy `Object`.
  - defaultne spravanie equals.
- Prekrytim je mozne definovat kedy su dva objekty navzajom rovne.
  - definuje relaciu rovnosti na objektoch. Musi byt reflexivna, symetricka a tranzitivna.
  - aka je to teda relacia ?
  - kontrakt metody (javadocky class `Object`)
- Ked prekryvame equals VZDY prekryvame aj metodu `hashCode`.

Priklad

Vytvorte vlak.

- Vytvorte triedy Locomotive (lokomotiva), PersonalCarriage (osobny vagon) a CargoCarriage (nakladny vagon). U kazdej z nich bude v konstruktore povinnost zadat oznacenie drah ako retazec.
- U vagonov sa tiez eviduje dlzka vagona(o lokomotivy nie) - zadava sa v konstruktore. U osobneho vagona sa eviduje pocet sedadiel a pocet ludi, ktory v nom prave su, u nakladneho maximalna nosnost a momentalne nalozenie.
- U lokomotivy sa eviduje jej vykon v kW.
- Vytvorte triedu Train. V konstruktore sa jej vzdy zada Lokomotiva. Podla jej vykonu sa urci kolko najviac vagonov moze vlak mat. Na jeden vagon musi mat lokomotiva vykon 50 kW ale najviac moze mat vlak 30 vagonov(nariadenie drah).
- Je mozne mat zmiesane vlaky (aj nakladne aj osobne vagony). Vlak bez lokomotivy moze mat najviac 500 metrov (nariadenie drah).

Priklad pokracovanie

- Vytvorte metody na vratenie poctu vagonov a dlzky vlaku.
- Vytvorte metodu na pridanie vagonu. Metoda zliha ak uz nie je mozne pridat vagony. Ak by pridanim vagonu vlak presiahol 500 metrov vyhodte vynimku TooLongTrainException, ktora rozsiruje TrainException.
- Vytvorte metodu na vratenie lokomotivy a x-teho vagonu. Ak x je vacsie ako momentalny pocet vagonov vlaku ale mensi ako potencialny mozny pocet vagonov vyhodte vynimku NoCarriage ktora rozsiruje TrainException.
- Vytvorte metodu ktora vrati pocet cestujucich vlaku a pocet kilogramov nakladu.
- Prekryte metodu equals u vsetkych tried: lokomotivy su rovnake ked maju rovnake oznacenie. Vagony su rovnake len z vagonami rovnakeho typu a to vtedy ked maju rovnake oznacenie. Vlaky su rovnake ked maju rovnaku lokomotivu a pocet vagonov[1].

[1] Trosku neprirodzeny priklad ale povedzme :)

Domaca uloha

- Uloha bude na dalsej hodine hned na zaciatku.
- Upravte priklad na zasobnik z davnejsich cviceni tak aby sa do zasobniku dali vkladat lubovolne objekty a nie len cisla(neimplementujte zasobnik pomocou pola !! Pole ma pevnu dlzku, zasobnik nie).
- vytvorte metodu removeCarriage ktora odobere posledny vagon vlaku. Pretazte rovnomennou metodou s parametrom typu int, ktora umozni odobrat zadany vagon vlaku.
- Vytvorte nakladny vlak a osobny vlak. Zmente implementaciu osobnych vlakov tak aby vlak bol identifikovany svojim oznacenim (ktore vidime napr. v cestovnom poriadku). 2 osobne vlaky rovnake ked maju rovnake oznacenie.
- Implemntujte metody toString vsetkych tried. Vypisu podrobne info. V pripade vlaku vypisu info o vlaku a potom postupne o vsekych jeho vagonoch.

Domaca uloha 2

- Vytvorte triedu Main v ktorej si vyskusate vytvorit vlak a niekolko vagonov ktore mu pridate. Co sa stane ked zamerne proridate moc dlhy vagon ? Vyskusajte. Co sa stane ked sa zamerne pokusite vybrat vagon ktory este nie je pridany ? Vyskusajte. Skuste vynimky osetrit tak aby sa na vystup vypisala chybova hlaska.
- Ponukam bonusovy bod za vytvorenie interaktivneho konzoloveho klienta na spravu vlaku. Bude mozne vytvorit vlak, zadat typ vlaku - univerzalny, nakladny, osobny, pridavat a odoberat vagony. Bude mozne zadat pocet ludi ktory pritupili ci vystupili, obdobne u nakladu. Taktiez nechat si vypisat pocet ludi vo vlaku prip. naklad. Je mozne ziskat informacie o celom vlaku alebo o zadanom vagone. Bude mozne ukoncit pracu z vlakom(zahodit ho) a vytvorit novy vlak.