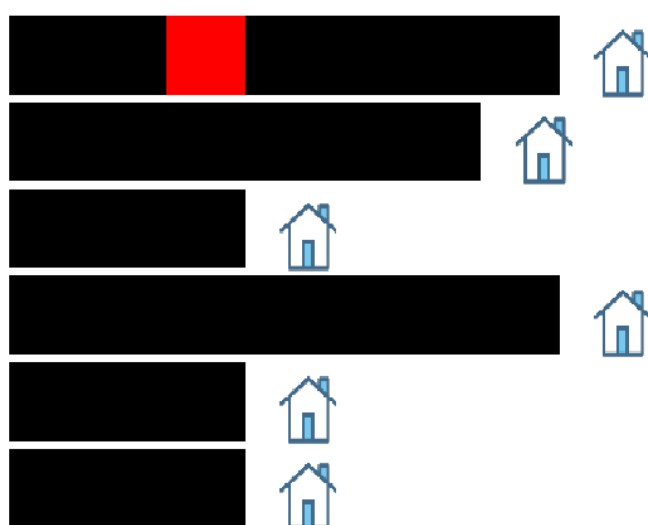


Zadanie 9. týždeň – návrat z baru pešo

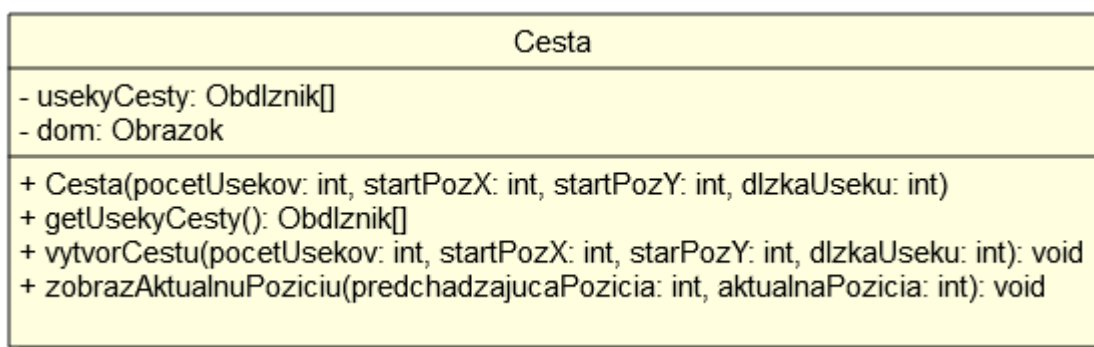
Minulý týždeň bola partia ľudí v bare, avšak pri ich návrate domov zastavil každé auto policajt. Vzhľadom na nezodpovednosť vodičov, všetci prišli o vodičské preukazy. To však nikoho nezastavilo prísť o týždeň do baru zas – avšak teraz musia ísť všetci domov pešo.

Už vieme, že osoby v bare čo-to popijú a našou úlohou bude ich z baru dostať úspešne domov. Problémom je, že niektoré osoby majú príliš veľa alkoholu v krvi na to, aby ich cesta bola priama a kým sa dostanú domov, spravia pár krokov navyše.

Na nižšie uvedenom obrázku vidíte cesty domov jednotlivých štangastov. Prvá osoba je práve v prvej polovici svojej strastiplnej cesty (aktuálna pozícia je zobrazená ako červený štvorček).



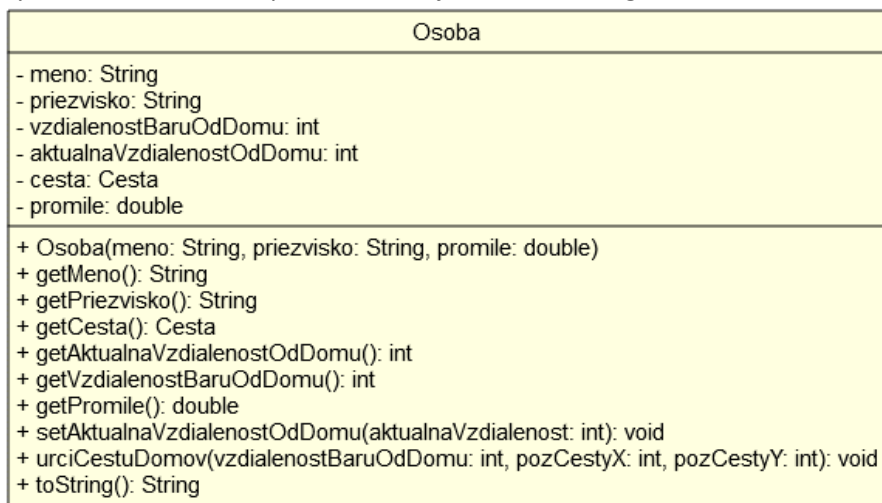
1. Vytvorte triedu *Cesta* podľa nižšie zobrazeného UML diagramu.



- Atribút *usekyCesty* bude pole obdĺžnikov predstavujúce cestu domov. Každá cesta má na svojom konci obrázok domu.
- Metóda *vytvorCestu* inicializuje pole *usekyCesty* na veľkosť *pocetUsekov*.
 - Pre každý úsek sa v cykle vytvorí nový obdĺžnik, ktorého strany budú *dlzkaUseku x dlzkaUseku*.
 - Polohu obdĺžnika treba určiť na základe poradia úseku v poli.

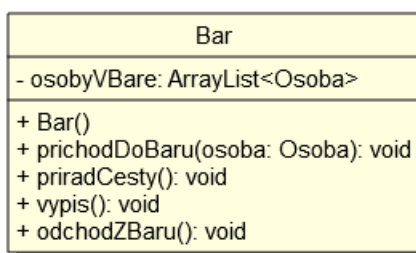
- Farbu každého obdĺžnika nastavte na čiernu.
- Vytvorený obdĺžnik uložte do poľa *usekyCesty*.
- V konštruktore zavolajte metódu *vytvorCestu* a vytvorte dom na konci cesty. Veľkosť domu nastavte na 70 x 50.
- Metóda *zobrazAktualnuPoziciu* zmení úseku cesty na predchádzajúcej pozícii farbu na čiernu (ale iba ak predchádzajúca pozícia nie je -1). Aktuálnemu úseku cesty zmeníte farbu na červenú.

2. Vytvorte triedu *Osoba* podľa nasledujúceho UML diagramu.



- V konštruktore nastavte atribúty *meno*, *priezvisko* a *promile*.
- V metóde *urciCestuDomov* nastavte atribúty *vzdialenostBaruOdDomu* a *aktualnaVzdialenostOdDomu* na hodnotu *vzdialenostBaruOdDomu*, ktorá príde ako parameter a predstavuje počet úsekov cesty. Vytvorte cestu.
- Metóda *toString* vypíše "*meno priezvisko má cestu domov vzdialenú vzdialenostBaruOdDomu km.*"

3. Vytvorte triedu *Bar* podľa UML diagramu.



- V metóde *prichodDoBaru* pridajte osobu do kontajnera *osobyVBare*.
- V metóde *vypis* zavolajte na konzolu vypíšte *toString* každej osoby v bare.
- Metóda *priradCesty* každej osobe v bare vygeneruje náhodnú dĺžku cesty v intervale od 1 po 10 úsekov. Následne sa každej osobe určí cesta domov (použite metódu triedy *osoba*, ktorá je na to určená!). Každá cesta bude začínať na X-pozícii rovnej 50. Cesta prvej osoby bude začínať na Y-pozícii rovnej 50 a každá ďalšia cesta bude posunutá o hodnotu 55 na Y osi.

Príklad vygenerovania náhodného celého čísla v intervale od 50 do 100:

```
Random r = new Random();
```

```
int nahodneCislo = r.nextInt(50) + 50;
```

- V metóde *odchodZBaru* sa môžete držať podrobného postupu napísaného o jednu čiernu bodku nižšie, alebo zvolíte vlastný postup, ktorého výsledkom musí byť daná funkcionálnosť:
 - Každá osoba bude kráčať po ceste, kým nepríde domov. Avšak, ak je osoba pod výrazným vplyvom alkoholu, nebude sa posúvať stále len čelom vpred, ale občas sa vráti aj vzad.
- V metóde *odchodZBaru* posúvajte každú osobu v bare kým nepríde domov.
 - Pre každú osobu najskôr nastavte jej predchádzajúcu pozíciu na -1 a aktuálnu pozíciu na hodnotu 0.
 - Kým osoba nie je doma:
 - Aktualizuje sa jej aktuálna vzdialenosť od domu
 - Ďalej zavolajte metódu `Platno.dajPlatno().wait(350);` ktorá chvíľu počká, kým sa zobrazí ďalší krok. Do triedy importujte knižnicu `java.util.concurrent.TimeUnit`
 - Zobrazte aktuálnu pozíciu osoby na ceste
 - Aktualizujte predchádzajúcu pozíciu
 - Posuňte aktuálnu pozíciu osoby na ceste:
 - Ak je aktuálna pozícia rôzna od 0 a desatina promile je viac ako náhodne vygenerované desatinné číslo:
 - posuňte osobu doľava
 - Inak
 - posuňte osobu doprava
 - Ak je osoba na konci svojej cesty, tak ukončíte cyklus kráčania domov po ceste
 - Oznámte, že osoba prišla domov

4. Stiahnite si triedu `Main.java` a pridajte ju do projektu.