



Záverečný test Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmýšľaj), raz rež (programuj)

Dôležité pravidlá a informácie (viac na stránke predmetu):

- čas na riešenie úloh je **240 minút**,
- nie je dovolená žiadna komunikácia s kýmkoľvek okrem dozoru,
- v prípade akýchkoľvek problémov alebo z dôvodu ohodnotenia riešenia kontaktujte dozor,
- riešenia je možné nechať si ohodnotiť aj priebežne (nie až v závere testu),
- **funkčnosť každej metódy musí byť preukázaná spustením na vami vytvorenom testovacom vstupe, nespustiteľné metódy neumožňujú zisk príslušných bodov,**
- všetky inštančné premenné musia byť neverejné.



PAZ Fekal s.r.o.

alebo najhnusnejšie zadanie v dejinách predmetu PAZ1a

Motivácia: Novozaložená firma PAZ Fekal s.r.o. sa rozhodla pomôcť obciam a prípadne vodárenským spoločnostiam porozumieť dátam o produkcii odpadových vôd. V tomto systéme sa spracuje evidencia domácností za uplynulý kalendárny rok a následná analýza porovnáva tieto údaje pre jednotlivé domácnosti. Domácnosť platí vodárenskej spoločnosti poplatky za spotrebu vody a produkciu odpadových vôd na základe cenníka vodného a stočného. V prípade vypúšťania odpadových vôd do verejnej kanalizácie sa objem týchto vôd vypočíta na základe meranej spotreby vody. Nie každá obec a každá domácnosť má však prístup k verejnej kanalizácii. V takom prípade je odpadová voda odvádzaná do žumpy a následne odvezená do príslušnej čistiarne odpadových vôd. Môžete predpokladať, že vývoz žumpy znamená jej úplné vyčerpanie a kapacita žumpy nepresiahne maximálnu kapacitu vozidla s cisternou (typicky 11m^3). Producent bez pripojenia na verejnú kanalizáciu neplatí stočné.

Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu *Domacnost*, ktorá uchováva informácie o jednom rodinnom dome alebo bytovej jednotke
- triedu *Obec*, ktorá bude uchovávať zoznam domácností.

Zadanie: V balíku `sk.upjs.fekal` vytvorte triedu *Domacnost* obsahujúcu dátové položky prístupné cez `gettre` (a podľa uváženia aj modifikovateľné cez `settre`):

- **meno** (meno a priezvisko obyvateľa domu, ktorý vystupuje ako odberateľ vody resp. producent odpadových vôd voči vodárenskej spoločnosti, resp. názov právnickej osoby),
- **adresa** (adresa domácnosti vo formáte “*ulica číslo domu, PSČ obec*“, napr. *Hlavná 12, 04001 Košice*. V prípade neexistencie ulíc je namiesto ulice uvedený názov obce, napr. *Kysak 100, 04481 Kysak*)
- **pocetObyvatelov** (počet členov domácnosti s trvalým pobytom na danej adrese, v prípade právnickej osoby je počet obyvateľov nula),
- **rocnaSpotreba** (množstvo spotrebovanej vody v m^3 za sledovaný kalendárny rok),
- **kanalizacia** (označuje, či má domácnosť pripojenie na verejnú kanalizáciu),
- **kapacitaZumpy** (označuje objem žumpy v m^3 , v prípade napojenia na verejnú kanalizáciu má kapacitaZumpy hodnotu 0m^3)
- **pocetVывozov** (označuje koľkokrát bola vyvezená žumpa za sledované obdobie),

Upozornenie: Zadanie pre triedu `Domacnost` predpisuje dátové položky prístupné cez `gettre`. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí. Napríklad nie je potrebná samostatná premenná pre označenie, či je domácnosť pripojená na kanalizáciu (dá sa to zistiť na základe uvedenej kapacity žumpy).

Kapacita žumpy je udávaná v m^3 – teda napr. $11m^3$ alebo $5.5m^3$. V prípade potreby v samotnej premennej alebo v príslušných metódach môžete pracovať aj s jednotkou liter ($1m^3=1000$ litrov). Môžete predpokladať, že všetky hodnoty v evidencii domácností sú zaokrúhlené na celé litre. V súbore je kapacita žumpy uvedená v m^3 .

Ďalej vytvorte triedu `sk.upjs.fekal.Obec`, ktorá bude uchovávať zoznam domácností.

Konštruktory a evidovanie domácnosti (3 body dokopy – povinné):

- **public** `Domacnost(String meno, String adresa, int pocetObyvatelov, double rocnaSpotreba)` – použije sa na evidovanie domácnosti pripojenej na verejnú kanalizáciu.
- **public** `Domacnost(String meno, String adresa, int pocetObyvatelov, double rocnaSpotreba, double kapacitaZumpy, int pocetVyvozov)` – použije sa na evidovanie domácnosti bez pripojenia na verejnú kanalizáciu.
- **public void** `eviduj(Domacnost domacnost)` – inštančná metóda v triede `Obec`, ktorá zaeviduje údaje o domácnosti za sledovaný kalendárny rok.

Práca so súbormi (povinné):

V triede `Domacnost`:

- **public static** `Domacnost zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Domacnost`. Parameter je reťazec v tvare `"meno\tadresa\tpocetObyvatelov\trocnaSpotreba"`, resp. `"meno\tadresa\tpocetObyvatelov\trocnaSpotreba\tkapacitaZumpy\tpocetVyvozov"` ak domácnosť nemá pripojenie na verejnú kanalizáciu (3 body);
Poznámka: Znak `\t` je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci údaje o domácnosti (1 bod).

V triede `Obec`:

- **public static** `Obec nactajEvidenciu(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta evidenciu domácnosti (zoznam domácností), pričom v každom riadku bude popis jednej domácnosti (4 body).
- **public void** `ulozEvidenciu(String nazovSuboru)` – uloží všetky zaevidované domácnosti obce do súboru v tvare, ktorý vie spracovať metóda `nactajEvidenciu(String nazovSuboru)` (3 body).
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci kompletnú evidenciu pre danú obec (1 bod).

Inštančné metódy triedy `Domacnost` (náročnejšie metódy, môžete ich riešiť neskôr):

- **public double** `vypocetPlatby(double stocne, double cenaZaVyvoz)` - vypočíta a vráti ročnú sumu poplatku za likvidáciu odpadových vôd. Suma stočného, ktorú platí producent pripojený na verejnú kanalizáciu je udávaná v EUR za $1 m^3$. Cena za vývoz zodpovedá vyčerpaniu a odvozu celého objemu žumpy bez ohľadu na množstvo (2 body).

- **public** String ulica() - vráti názov ulice prislúchajúcej danej domácnosti. Vráti **null** v prípade, že obec nemá vyznačené ulice a teda adresa začína aj končí názvom obce (3 body + 3 body za verziu, ktorá funguje pre viacslovné názvy ulíc a obcí).
- Nech trieda Domacnost implementuje rozhranie Comparable<Domacnost>. Doplňte metódu **public int** compareTo(Domacnost d) tak, aby sa porovnávali domácnosti podľa ročnej spotreby vody v prepočte na jedného obyvateľa. V prípade právnických osôb sa berie do úvahy celková spotreba vody. Podľa metódy je *menšia domácnosť* tá s menšou prepočítanou spotrebou (4 body).

Inštančné metódy triedy Obec (úlohy môžete riešiť v ľubovoľnom poradí):

Ak niektorá z metód nevie vrátiť referenciu na objekt s požadovanými vlastnosťami, metóda nech vráti **null**.

- **public double** pokrytie() - vráti aké percento obyvateľov obce má prístup k verejnej kanalizácii (2 body).
- **public boolean** kompletnaEvidencia(int pocetObyvatelov) - podľa počtu obyvateľov obce overí, či boli získané údaje o všetkých domácnostiach (2 bod).
- **public int** kapacitaCisterny() - vodárne disponujú rôznymi vozidlami (typicky s kapacitou 1 l m³ a 5 m³). Obec sa rozhodla zakúpiť vlastné vozidlo s cisternou. Metóda vráti akú najmenšiu kapacitu v litroch si môže dovoliť, aby bola schopná na jeden vývoz vyčerpať akúkoľvek žumpu v obci (2 body).
- **public** List<Domacnost> aktivneDomacnosti() - vráti referenciu na list domácností, kde budú uvedené iba tie domácnosti z obce, kde je odberateľom fyzická osoba (na danej adrese je nahlásená aspoň jedna osoba na trvalý pobyt) a za uplynulý rok bolo spotrebovaných aspoň 10 m³ vody alebo bola aspoň trikrát vyvezená žumpa (2 body).
- **public** Domacnost najvacsieOdberneMiesto() - vráti referenciu na domácnosť s najväčším odberom vody v rámci celej obce (2 body).
- **public double** celkovaProdukcia(String ulica, int odCislaDomu, int poCisloDomu) - vráti celkovú produkciu odpadových vôd (určenú na základe spotrebovanej vody) na zadanej ulici od zadaného čísla (odCislaDomu) po iné číslo (poCisloDomu). Môžete predpokladať, že čísla domov idú v obci v správnom poradí. Odporúčame vytvoriť si pomocnú metódu v triede Domacnost, ktorá na základe svojej adresy vráti číslo domu ako **int** (6 bodov).
- **public** List<String> podozrivaLikvidacia() - vráti zoznam mien odberateľov u ktorých je podozrenie, že vypúšťajú obsah žump do prostredia (napr. do pôdy alebo do potoka). Podozrivou domácnosťou je taká, ktorej celkový objem vyvezených odpadových vôd sa líši od spotreby vody o viac ako 10% z ročného objemu spotrebovanej vody (4 body).
- **public** List<String> setriaceRodiny() - obec sa rozhodla odmeniť 5 rodín s najnižšou celkovou spotrebou vody za domácnosť. Metóda vráti usporiadaný zoznam maximálne 5 mien odberateľov, ktorí reprezentujú domácnosti s aspoň 4 členmi a majú najnižšiu ročnú spotrebu za domácnosť v rámci obce. V prípade potreby implementujte rozhranie Comparator<Domacnost> (7 bodov).
- **public** Map<Double, Integer> databazaZump() - vráti počty žump s jednotlivou kapacitou (3 body).
- **public double** usetreneProstriedky(String ulica, double stocne, double cenaZaVyvoz) - vráti očakávané množstvo peňazí ušetrených obyvateľmi obce v prípade rovnakej spotreby vody v nasledovnom období za predpokladu vybudovania kanalizácie na zadanej ulici (3 body).
- **public void** prepisOdberatelov(Map<String, String> prepisy) - v evidencii obce sú mená odberateľov vody podľa údajov zo začiatku sledovaného obdobia. V mape referencovanej premennou prepisy sú uvedené všetky zmeny zaznamenané vodárenskou spoločnosťou počas roka. Kľúčom v mape je pôvodný odberateľ a hodnotou nový odberateľ. Metóda prepisOdberatelov aktualizuje údaje v evidencii. Za každú domácnosť môžete predpokladať maximálne jeden prepis za aktuálny rok (3 body + 2 body ak riešenie bude prechádzať zoznamom domácností iba raz).

- **public** Set<String> velkeFirmy() - vráti množinu obchodných názvov firiem, ktoré majú viac odberných miest v rámci obce (4 body).
- **public** List<Domacnost> domacnostiPodlaOdberu() - vráti referenciu na novovytvorený zoznam všetkých evidovaných domácností usporiadaný podľa spotreby vody v prepočte na obyvateľa od najväčšieho po najmenší (4 body).
- **public** Map<String, Double> vyuctovanie(double stocne, double cenaZaVyvoz) - vráti referenciu na objekt triedy implementujúcej rozhranie Map, kde kľúčom bude meno odberateľa a hodnotou súčet platieb za všetky domácnosti, kde je registrovaný ako odberateľ. Cena sa vypočíta rovnakým spôsobom ako pri metóde vypocetPlatby v triede Domacnost (4 body).

Výnimka a modifikátory:

Pridajte do triedy Obec inštančnú premennú nazovObce, ktorá uchováva oficiálny názov obce. Jej hodnota nech sa nastavuje v konštruktore a nebude ju možné zmeniť. Premenná nesmie byť public, ale nech je priamo dostupná v akejkoľvek triede z balíčka sk.upjs.fekal (2 body)

Vytvorte nekontrolovanú výnimku NeplatnyNazovObceException a vhodne ju použite pri evidovaní novej domácnosti v prípade, že sa názov obce v adrese pridávanej domácnosti nezhoduje s oficiálnym názvom obce (3 body).