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 (elektronická aj neelektronická) 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.

DOD na ÚINF-e

zdroj: https://www.upjs.sk/

Motivácia: Určite viete, že 2.2. bude deň otvorených dverí na PF UPJŠ. Čo je naša najväčšia akcia pre stredoškolákov. Preto vás chceme poprosiť aby ste pozvali svojich kamarátov zo stredných škôl na naše DOD. Ale dnes sa pozrieme na organizáciu tohto a ďalších podujatí z druhej strany. Po určení termínu a miesta konania je dôležite si naplánovať, čo chceme prezentovať, akým štýlom, kto si vezme ktorú časť a čo na to potrebujeme. Aj keď podobných akcii je niekoľko do roka a robíme to už mnoho rokov, stále sa držíme hesla „If you fail to plan you plan to fail“. Tak naplánuj svojim kamarátom DOD u nás a nám pomôž spraviť program na plánovanie DODčka a ďalších podujatí.

Pohľad analytika: Pri implementácii aplikácie budeme potrebovať:

- triedu `Prezentacia`, ktorá reprezentuje jednu prezentáciu na DOD alebo inom podujatí,
- triedu `PlanPodujatia`, ktorá bude uchovávať zoznam všetkých prezentácií v ponuke.

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

- **`nazov`** (názov prezentácie napr. Pomocník náhoda, Automaty v našich vreckách alebo iné)
- **druh** (napr. prednáška, workshop, stánok, diskusia alebo iné)
- **dlzka** (dĺžka v minútach)
- **predchadzajucePodujatie** (ak ide o prezentáciu ktorá bola využitá aj v minulosti, tak je uvedený názov posledného podujatia, kde bola táto prezentácia napr. „DOD PF UPJŠ 2022“, „Noc výskumníkov 2022“)
- **hodnotenie** (ak ide o prezentáciu ktorá bola využitá aj v minulosti, tak zo spätnej väzby má spočítané priemerné hodnotenie, číslo medzi 0 až 10, napr. 9.2 alebo 7.9, hodnotenie môže byť aj nevyplnené ak ide o novú prezentáciu)
- **zodpovednaOsoba** (meno osoby zodpovednej za prezentáciu napr. Jožko Mrkvička)
- **technickeZabezpecenie** (zoznam technického a materiálneho zabezpečenia oddeleného čiarkami napr. „počítačová učebňa“ alebo „stôl, počítač, rollup“ alebo „oculus, prezenter, notebook, miestnosť s voľným priestorom“ alebo iné)

Upozornenie: Zadanie pre triedu `Prezentacia` predpisuje dátové položky prístupné cez gettre. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.PlanPodujatia`, ktorá bude uchovávať zoznam prezentácií.

Konštruktory a pridávanie prezentácií do plánu podujatia (povinné):

- `public Prezentacia(String nazov, String druh, int dlzka, String predchadzajucePodujatie, double hodnotenie, String zodpovednaOsoba, String technickeZabezpecenie)` – použije sa na vytvorenie prezentácie, ktorá už bola v minulých rokoch (s vyplneným hodnotením).
- `public Prezentacia(String nazov, String druh, int dlzka, String zodpovednaOsoba, String technickeZabezpecenie)` – použije sa na vytvorenie novej prezentácie (bez hodnotenia).
- `public void pridaj(Prezentacia prezentacia)` – inštančná metóda v triede `PlanPodujatia`, ktorá pridá prezentáciu do plánu podujatia.

Práca s reťazcami a súbormi (povinné):

V triede `Prezentacia`:

- `public static Prezentacia zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Prezentacia`. Parameter je `String` v tvare "`nazov \t druh \t dlzka \t predchadzajucePodujatie \t hodnotenie \t zodpovednaOsoba \t technickeZabezpecenie`", resp. "`nazov \t druh \t dlzka \t zodpovednaOsoba \t technickeZabezpecenie`", ak ide o prezentáciu ktorá ešte nebola na žiadnom predchádzajúcom podujatí.
  Poznámka: Znak `\t` je neviditeľný znak tabulátora. `Scanner`-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Medzera pred a za `\t` sú len kvôli zlepšeniu čitateľnosti zadania, v reťazci reálne nie sú.
- `public String toString()` – vráti reťazec vhodne reprezentujúci údaje o prezentácii.

V triede `PlanPodujatia`:

- `public static PlanPodujatia zoSuboru(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta plán podujatia, pričom v každom riadku bude popis jednej prezentácie.
- `public void uloz(String nazovSuboru)` – uloží všetky prezentácie zo zoznamu prezentácií do súboru v tvare, ktorý vie spracovať metóda `zoSuboru(String nazovSuboru)`.
- `public String toString()` – vráti reťazec vhodne reprezentujúci všetky prezentácie uchované v zozname aktivít.

Za povinnú časť bude udelených 15 bodov. Nasledujúce (nepovinné) úlohy môžete riešiť v ľubovoľnom poradí. Ale riešenie jednej môže zjednodušiť nasledujúce.

Inštančné metódy triedy `plán podujatia`:

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

- `public int vratCas()` – vráti na koľko minút sú plánované dokopy všetky prezentácie (1 bod).
- `public int najdlhsiaPrezentacia()` – vráti koľko minút trvá najdlhšia prezentácia (1 bod).
- `public List<Prezentacia> novePrezentacie()` – vráti zoznam prezentácií, ktoré nemajú vyplnené hodnotenie, teda neboli na žiadnom predchádzajúcom podujatí (1 bod).
- `public List<String> vratPrezentacieZPodujatia(String nazovPredchadzajucehoPodujatia)` – vráti názvy všetkých prezentácií, ktoré majú uvedené predchádzajúce podujatie rovnaké ako zadané parametrom, vo vrátenom zozname sa môže každý názov vyskytovať najviac raz (2 body).
- `public List<String> nepopularnePrezentacie(int hodnotenie)` – vráti zoznam názvov prezentácií ktoré majú hodnotenie menšie rovné ako je zadané parametrom (2 bod).
- `public boolean prednasatel(String zodpovednaOsoba)` – metóda vráti `true` ak na každej prezentácii kde je daná zodpovedná osoba je vždy napísaný druh „prednáška“ (2 body).

- `public Map<String, Double> percentoDruhu()` – vráti mapu, v ktorej je ku každému druhu prezentácie priradený percentuálny podiel času z celkového času podujatia. (3 body). +1 bod za zaokrúhlenie na 2 desatinné miesta.
- `public Map<String, Integer> zoznamTechnickehoZabezpecenia()` – vráti mapu technického zabezpečenia, kde je každému názvu priradený počet koľkokrát je uvedený v prezentáciách (4 body). +4 body za „smart“ verziu, kde každej zodpovednej osobe stačí priradiť zariadenie raz. Napr. ak Jožko potrebuje notebook na tri prezentácie a Majka iba raz, tak stačia dva notebooky (každému jeden).
- `public List<String> neinovativneOsoby()` – vráti zoznam tých zodpovedných osôb, ktorých všetky prezentácie už boli v minulosti použité (majú hodnotenie) (4 body).
- `public boolean podobnyPlan()` – vráti `true` iba ak aspoň polovica času prezentácií je z prezentácií, ktoré majú rovnaké predchádzajúce podujatie (4 bodov).
- `public int[] minutazVKvalite()` – vráti pole veľkosti 10, každý index zodpovedá kvalite napr. index 0 kvalite <0,1), index 1 kvalite <1,2) atď. až index 8 kvalite <8,9) a index 9 kvalite <9,10>. Vo vrátenom poli sa na indexe i nachádza koľko minút dokopy je v prezentáciách s daným hodnotením (5 bodov).
- `public boolean pridajMinulePodujatie(PlanPodujatia starePodujatie, String nazovStarehoPodujatia)` – metóda pridá do aktuálneho plánu všetky prezentácie zo starého podujatia a navyše im nastaví prechádzajúce podujatie podľa jeho názvu. Metóda vráti `true` ak názov žiadnej prezentácie nebol už pred pridaním v aktuálnom pláne podujatia (5 bodov). +1 bod za „čistotu kódu“.
- `public String pripravProgram(int cas, String zaciatok)` – metóda vráti reťazec pre program podujatia s minutážou, pričom nenaplnený čas je rovnomerne rozdelený do prestávok. Reťazec má uvedený názov prezentácie a doby trvania, každá prezentácia je v novom riadku. (7 bodov). *Príklad:* Pre 3 prezentácie s názvami P1, P2, P3, každá s dĺžkou 20 minút, čas 80 minút a začiatok o 9:10 metóda vráti „9:10-9:30 P1 \n 9:40-10:00 P2 \n 10:10-10:30 P3“, znak „\n“ vyjadruje nový riadok a medzery okolo neho sú iba kvôli čitateľnosti podobne ako v úlohe pri práci so súbormi.
  Pozn.: Žiadne dve prezentácie neprebiehajú súbežne. Čas je aspoň taký ako vráti metóda `vratCas`.
- `public void topPrezentacie()` – metóda vypíše prezentácie podľa ich hodnotenia od najvyššieho po najnižšie potom nasledujú prezentácie bez hodnotenia zoradené lexikograficky (podľa abecedy) (6 bodov). +2 až 3 body (podľa „čistoty kódu“) za pridanie preťaženej metódy s parametrom `int n` a výpis iba prvých `n` názvov prezentácii.

Nezaradené úlohy

Vytvorte a použite statickú nemennú premennú, ktorá bude reprezentovať prezentácie na 0 minút napr. postavený rollup, nástenka, samoobslužný stánok s občerstvením. (1 bod)

Pridajte nekontrolovanú výnimku HodnotenieMimoRozsahuException, ktorá sa bude vhodne týkať prípadu, ak je hodnotenie mimo rozsahu <0,10>. Výnimku použite na vhodnom mieste (2 body).