

1. Domácí úloha 06

Základní informace:

- **Účel:** konstrukce hlavní třídy, využití Služebníka pro další třídu
- **Kostra:** 06_HlavniTrida.zip
- **Odevzdávaný soubor/JAR:** 06_HlavniTrida.jar
- **Odevzdávaný soubor UML:** 06_A11B0987P.png - každý použije své osobní číslo

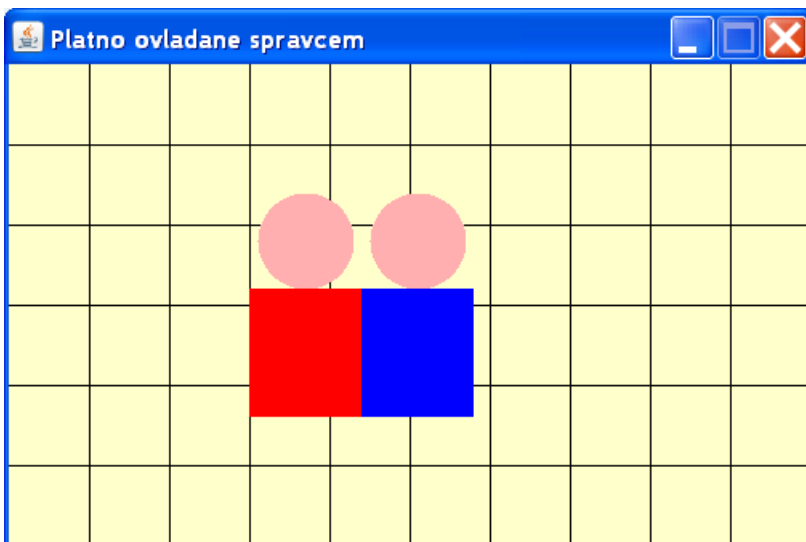
Zadání:

- připravte třídu `Rande`
- připravte třídu `Par`
- připravte třídu `Hlavni`
- do Portálu odevzdáte JAR soubor celého projektu
- rozšířte UML diagram tříd

Postup řešení:

- stáhněte si soubor 06_HlavniTrida.zip, rozbalte jej - NEotvírejte projekt v BlueJ
- do rozbaleného adresáře nakopírujte soubory `Osoba.java`, `Rozmer.java`, `Pohlavi.java`, `IMeritelny.java`, `IZvyrazneny.java` a `Zvyraznovac.java`, které jste odevzdávali v minulém DU
- v BlueJ otevřete projekt 06_HlavniTrida
- připravte třídu `Rande` tak, že bude schopna vytvořit schůzku muže a ženy
 - deklaruje a inicializuje pomocnou konstantu třídy `SpravcePlatna` se jménem `SP`
 - deklaruje konstanty `muz` a `zena`
 - deklaruje konstanty `domaMuz` a `domaZena`, které budou uchovávat počáteční (domovskou) pozici muže a ženy
 - připravte konstruktor se signaturou `public Rande(Pozice domaMuz, Pozice domaZena)`
 - ◆ konstruktor vytvoří na zadaných pozicích:
 - pomocí statické tovární metody `muze`
 - pomocí nejvhodnějšího konstruktoru `Osoba` ženu
 - toto odlišné vytváření nemá praktický význam, je použito jen pro ukázkou možnosti vzniku instance více způsoby
 - ◆ nezapomeňte zadané pozice uložit do příslušných proměnných pro pozdější použití
 - ◆ konstruktor zajistí zobrazení vytvořených osob v domovských pozicích

- připravte getry pro atributy `muz`, `zena`, `domaMuz` a `domaZena`
- správnou funkci implementovaného konstrukturu a getrů ověřte pomocí *Testovat Vzniku*, kterou před prvním použitím odkomentujte
- připravte metodu se signaturou `public void jdouNaRande(Pozice mistoSchuzky, Presouvac chuzeNaRande)`:
 - ♦ pozor na skutečnost, že musí být dodržen bontón, takže žena jde vždy vpravo od muže (z pohledu muže) - na plátně je tedy vlevo
 - `mistoSchuzky` je tedy souřadnicí ženy
 - ♦ pár se dotýká (využijte služby získání šířky osoby), ale nepřekrývá se (zatím ;-)
 - ♦ muž jde samozřejmě na schůzku jako první a čeká tam na ženu
 - pomocí dodaného přesouvače přesuňte muže na místo schůzky páru - využijte metodu `presunO()` třídy `Presouvac`
 - muž přichází rovnou na své místo, které ale není přesně totožné s místem schůzky
 - souřadnice tohoto místa je třeba dopředu vypočítat - je posunuté o šířku ženy
 - ♦ přesune ženu na místo schůzky páru - využijte metodu `presunNa()` třídy `Presouvac`
 - ♦ protože jsou zde muž a žena stejně vysokí, stojí ve stejné výšce (viz obrázek)
 - ♦ je zajištěno, že `mistoSchuzky` je zvoleno na plátně tak, že se na něj pár vždy vejde, což nemusíte nijak testovat
- správnou funkci implementované metody ověřte pomocí *Testovat SetkaniUprostred*, *Testovat SetkaniVpravo* a *Testovat SetkaniVlevo*, které před prvním použitím odkomentujte



- připravte třídu `Par` tak, že bude schopna přesouvat daný pár s využitím již známého služebníka `Presouvac`
- deklarujte a inicializujte pomocnou konstantu třídy `SpravcePlatna` se jménem `SP`
- deklarujte privátní konstanty `muz` a `zena`

- připravte konstruktor se signaturou `public Par(Rande rande)`
 - ◆ konstruktor získá z `rande` odkazy na muže a ženu a uloží je do svých konstant:
 - ◆ třída `Par` dále implementuje rozhraní `IKresleny` a `IPosuvny`
 - u implementované metody `nakresli()` uložte kreslení muže a ženy do bloku, aby nedocházelo k blikání

```

    SP.nekresli(); {
    ...
    } SP.vratKresli();

```

podrobnosti viz [OOP-03 str.16]

- u třídy `Par` se - stejně jako u všech ostatních objektů - pracuje s levým horním rohem ohraničujícího obdélníka, tj. v této DU s pozicí ženy

■ doplňte třídu `Rande`

- připravte metodu se signaturou `public Par parJdeSpolecne(Pozice mistoVyletu, Presouvac chuzeSpolu)`, která:
 - ◆ vytvoří instanci třídy `Par` se skutečným parametrem odkazu na aktuální instanci `Rande`, tj. `this`
 - ◆ zaregistruje vzniklou instanci `par` u správce plátna
 - ◆ pomocí předaného přesouvače dopraví pár na `mistoVyletu`
 - ◆ správnou funkci implementované metody ověřte pomocí *Testovat JdouSpolu*, kterou před prvním použitím odkomentujte
- připravte metodu se signaturou `public void parPokracujeSpolecne(Par par, Pozice dalsiMistoVyletu, Presouvac chuzeSpolu)`, která:
 - ◆ pomocí předaného přesouvače dopraví `par` na `dalsiMistoVyletu`
 - ◆ správnou funkci implementované metody ověřte pomocí *Testovat PokracujiSpolu*, kterou před prvním použitím odkomentujte
- připravte metodu se signaturou `public void jdouDomu(Par par, Presouvac chuzeDomu)`, která:
 - ◆ pomocí předaného přesouvače dopraví `par` na domácí pozici ženy (muž doprovází ženu do jejího domova)
 - ◆ z domácí pozice ženy přesune muže z na jeho domácí pozici
 - ◆ správnou funkci implementované metody ověřte pomocí *Testovat CeleRande*, kterou před prvním použitím odkomentujte

■ připravte třídu `Hlavni` tak, že připraví scénář celé schůzky tak, aby proběhl podobně jako v *Testovat CeleRande*

- parametry metody `main()` budou tři celá čísla ve významu `rychlostNaRande`, `rychlostSpolu`, `rychlostDomu`

- ◆ parametry budou zadány vždy tři a vždy správně, což není třeba testovat
 - ◆ instance přesouvačů budou vytvořeny s odpovídajícími rychlostmi
 - ◆ v průběhu ani po skončení celé schůzky se nebudou vypisovat žádné potvrzovací zprávy pro uživatele
 - ◆ správnou funkci implementované metody ověřte pomocí *Testovat Hlavni*, kterou před prvním použitím odkomentujte
 - ◆ dále spusťte metodu `main()` tak, že z BlueJ pošlete zprávu `main()` třídě `Hlavni` - nezapomeňte nastavit parametry jako řetězce
- všechny vytvořené a upravované třídy prověřte pomocí PMD a odstraňte případné problémy
 - celý projekt již známým způsobem zabalte do JAR souboru `06_HlavniTrida.jar`, který budete odevzdávat
 - spusťte tento program několikrát z příkazové řádky a zkuste nastavit různé rychlosti
 - rozšiřte UML diagram tříd z minulého DÚ o nové třídy `Rande` a `Par` (třídu `Hlavni` nekreslete)
 - pravděpodobně budete muset změnit rozmístění některých minule nakreslených tříd
 - nahradte původní asociační vazby vedoucí ke všem implementovaným rozhraním přesnějším typem vazby
 - ◆ Tip: nemusíte mazat starou vazbu, stačí ji vybrat a v pravém boxu textově upravit
 - výsledek uložte a také exportujte jako PNG soubor `06_A11B0987P.png` - každý samozřejmě použije své osobní číslo
 - ◆ tento soubor budete odevzdávat do **Blok 12-OOP-UML**