

Test PR2

[PR2CT0915]

Vytvořte v NetBeans nový projekt pro test.

Dodržujte konvence psaní názvů tříd, metod, proměnných atd. Nezapomeňte členit program na části (podprogramy/metody)

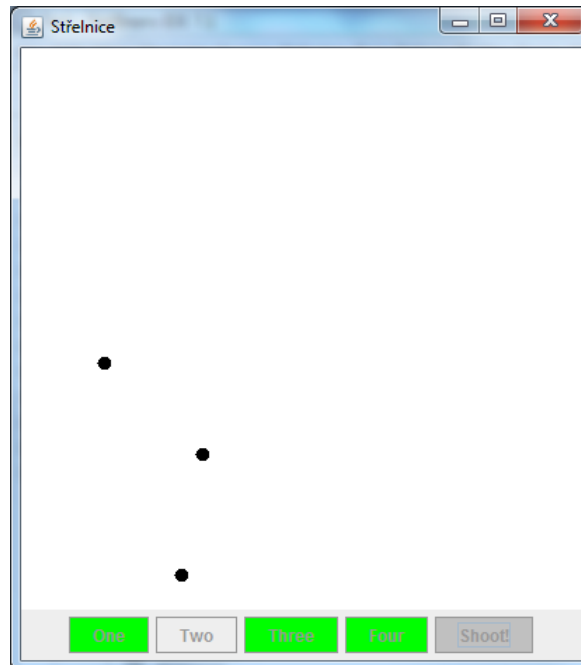
Odevzdávejte emailem s předmětem: „[PR2CT0915] test u PC“. Příloha bude ve formátu zip a pojmenována Username. ".zip" (příkaz „zip“, parametr -r ... rekurzivně)

Příklad:

- Vytvořte třídu DelovaKoule. Při vytváření bude nutné zadat **id** (řetězec), **počáteční rychlost ve směrech x a y** (reálné číslo). Rychlost nesmí být záporná, což bude signalizováno vhodnou výjimkou. Dále bude z objektu možné zjistit **souřadnice středu koule** (kvůli lepší čitelnosti a z důvodu nutné synchronizace si napište pomocnou třídu). Instance budou porovnávány pouze podle id. Objekt bude neměnný.
- U třídy implementujte rozhraní Runnable. Dokud bude souřadnice středu Y kladná (inicializujte ji např. na 1), volejte každých 50ms přepočítání podle následujících vzorců:
$$v_y = v_y - g \cdot dt$$
$$x = x + v_x \cdot dt$$
$$y = y + v_y \cdot dt - \frac{1}{2} g \cdot t^2$$
$$g = 9.8$$
$$dt = 0.1$$
- Zajistěte synchronizaci vláken. Nesmí se stát, že by bylo možné získat souřadnice nebo rychlost přepočítané napůl, nebo aby k přepočtu došlo mezi čtením jednotlivých souřadnic (napište si pomocnou třídu)
- Vytvořte třídu MnozinaKouli. Třída bude zapouzdřovat kolekci, která v Javě reprezentuje množinu. Použijte generiku k omezení typu prvku na DelovaKoule. Použijte návrhový vzor singleton. Pomocí metody Collections.synchronized...() zajistěte vláknovou bezpečnost množiny. (za ... doplňte)
- Vytvořte metodu pro přidání koule do kolekce. V případě, že daná koule již v množině existuje, signalizujte chybu. Pro ošetření chyby si napište vlastní třídu **asynchronní** výjimky. K detekci využijte vlastností kolekce a tříd (jediný příkaz!). Uvnitř výjimky předávejte zprávu o **id koule a důvodu chyby**.
- Třída MnožinaKouli bude dále implementovat rozhraní Iterable<DelovaKoule> čehož využijete při procházení prvků. Další veřejné metody budou **vyprázdnění kolekce** a **zjištění zda je prázdná**.
- Vytvořte třídu PaintPanel o rozměrech 400x400 (preferred size). Přepište zděděné metody `public void paint(Graphics g)` a `public void update(Graphics g)`. Metoda paint volá update. Update smaže obdélník o rozměru panelu (clearRect) a pro každý prvek v množině MnožinaKouli **vykreslí ovál** (fillOval) o šířce a výšce 10,10

na vhodné souřadnice. Zároveň informace o id a souřadnicích **vypište na konzoli**.

- Implementujte rozhraní Runnable a dokud MnožinaKoulí nebude prázdná každých 100ms překreslete panel (repaint).
- Bez designeru vytvořte třídu GUIPanel podle následujícího obrázku:



- Čtyři z tlačítek do MnožinaKoulí přidají novou kouli s id dle svého textu. Rychlost koule bude generována metodou getRandom, která vrátí náhodné číslo v intervalu pro směr x <10,50) a pro směr y <10,100). **V případě chyby** duplicitní koule zobrazí dialog (JOptionPane.show...) s titulkem „Duplicitní koule“ a zprávou předanou výjimkou. **V případě úspěšného přidání** změni pozadí tlačítka na zelené (setBackground) a pozadí posledního tlačítka na červené.
- Tlačítko Shoot při stisku spustí vlákno vykonávající posloupnost činností:
 - Nastaví své pozadí na LIGHT_GREY
 - Zneaktivní všechna tlačítka (setEnabled)
 - Pro každou kouli v MnožinaKoulí vytvoří a spustí vlákno
 - Spustí vlákno PaintPanel
 - Čeká, až všechna vlákna koulí skončí (musíme si pamatovat jaká jsme spustili)
 - Vyprázdní množinu koulí (na tom nám závisí vykreslování)
 - Povolí všechna tlačítka a zruší jejich pozadí (nastaví na null)
- Vytvořte třídu Test s metodou main. V té vytvořte okno, do něhož umístíte GUIPanel. Nezapomeňte na titulek a správné zavření aplikace.