

Členění kódu

Členění kódu, menší jednotky – jednodušší kompetence. Strukturovaný/hierarchický návrh. Vyhýbáme se opakování kódu. Návrh a použití vlastních metod. Více tříd v programu, knihovní třída.

Úlohy

1. Implementujte samostatné metody pro některé z dříve realizovaných výpočtů – faktoriál, mocnina, test přestupného roku, test rodného čísla apod.
2. Zapište program pro (již dříve uvedenou úlohu) *opakovaný výpočet úsekové rychlosti vozidel* tentokrát i s vyhodnocením závažnosti případného přestupku a stanovením výše případné sankce. Vytvořte program komunikující formou řádkového menu s položkami pro zadání parametrů úseku (délka, kde se nachází, povolená rychlost apod.), zadání údajů průjezdu konkrétního vozidla, vyhodnocení průměrné rychlosti, vyhodnocení případných sankcí. Metody: převod časového údaje na setiny sekund, výpočet průměrné rychlosti, výpočet výše pokuty a bodů (dle kritérií).
3. V jednom z předchozích zadání bylo zadáno vytvořit program, který *„bude realizovat hru „Hádání čísla“ a to tak, aby si uživatel mohl zvolit rozsah „myšlených/hádaných čísel“ a roli ve které se bude hry účastnit. Při jednom spuštění programu umožněte hraní libovolného počtu her (dle volby uživatele). Program má při jednotlivé hře počítat počet pokusů. Program má s uživatelem komunikovat formou řádkového menu. Program by mohl svoji činnost vykonávat následujícím způsobem – vypíše jednoduché menu, uživatel si zvolí akci, nastavení rozsahu čísel, hraní hry v roli prvního hráče, hraní hry v roli druhého hráče nebo ukončení hry, pokud uživatel zvolil akci, akce se provede (např. je zahájena a realizována příslušná hra do ukončení), poté program opět vypíše menu – atd.“* Pro realizaci tohoto zadání navrhnete vhodné členění programu do metod a implementujte.
4. Vytvořte knihovní třídu `VectorTools` pro manipulaci s 3D vektory – s metodami pro délku vektoru, skalární součin, úhel, test rovnoběžnosti, test kolmosti dvou vektorů, délku vektorového součinu (tj. délku vektoru, který je výsledkem vektorového součinu dvou vektorů). Kód metod/testů zapište bez použití logického příkazu. Při testování nezapomeňte na nepřesné zobrazení reálných čísel.. Metodám doplňte dokumentační komentáře. Vytvořte program, kterým otestujete implementované metody.
5. Vytvořte knihovní třídu `DateTimeTools` s metodami pro:
test přestupného roku;
test platnosti zadaného data;
test platnosti zadaného času;
výpočet dne týdne pro zadaný den (použijte Zellerův algoritmus);
určení počtu dní zadaného měsíce v konkrétním roce (výsledek 28, 29, 30 nebo 31);
určení pořadí dne v rámci roku pro zadané datum;
určení čísla týdne pro zadané datum;
převod zadaného data a času na jedinou hodnotu v definovaném formátu (v různých variantách převod na textový řetězec, celé číslo; prostředky pro obrácený převod.
Metodám doplňte dokumentační komentáře.
Vytvořte program, kterým otestujete implementované metody.
6. Vytvořte knihovní třídu `TriangleTools` s metodami pro: výpočet plochy obvodu a dalších parametrů (úhly, délky výšek, tečen, poloměrů kružnic apod.) trojúhelníka. Metodám doplňte dokumentační komentáře. Vytvořte program, kterým otestujete implementované metody.
7. Vytvořte knihovní třídu `NaturalNumbersTools` s metodami pro různé manipulace s přirozenými čísly (viz úlohy pro téma „Příkazy“ – počet dělitelů čísla, test prvočíselnosti, největší společný dělitel dvou čísel, nejmenší společná násobek dvou čísel, faktoriál at.). Metodám doplňte dokumentační komentáře. Vytvořte program, kterým otestujete implementované metody.

8. Vytvořte vlastní knihovnu třídu (třídu s metodami pro výpočet některých matematických funkcí) `MathTools`, ve které budou metody pro výpočet mocnin (celočíslné mocniny celých a kladných čísel, reálná mocnina čísla) a matematických funkcí uvedených v následujících dvou úlohách (výpočty `sin`, `cos`, `log`, `exp`, `sqrt`, `hypot` ...). Jednotlivým metodám doplňte dokumentační komentáře. Postupně realizujte program/programy, který otestuje implementované metody a umožní porovnání výsledků implementovaných metod s metodami ve třídě `Math`.
9. Doplňte třídu `MathTools` o metody pro výpočet některých z níže uvedených matematických funkcí pro zadané x jako součtu řady. Výpočet proveďte se zadanou přesností. Hodnota x a přesnost výpočtu `eps` budou zadávány jako parametry metody. Využití parametru `eps` v jednotlivých algoritmech – výpočet součtu ukončit v případě, že další člen řady je menší než požadovaná přesnost. Parametr `eps` v těchto případech představuje informaci „na kolik platných desetinných míst má být výsledek vypočten“.

Následně doplňte obdobné metody s jedním parametrem x , tyto metody mají provést výpočet příslušné hodnoty (součtu řady) s „absolutní přesností“ (přesností na plný počet platných číslic odpovídajících možností typu `double`). Výpočet ukončit v případě, že přičtením dalšího členu nedošlo ke změně výsledné hodnoty.

Vytvořte testovací kód (program) ve kterém otestujete jednotlivé metody. Výsledek metody vždy porovnejte s hodnotou vypočtenou voláním odpovídající metody třídy `Math` jazyka Java.

Přesnost výpočtu: konstantní hodnota zavedená v programu nebo (pro testování vhodněji) proměnná, jejíž hodnota bude zadaná uživatelem – přesnost musí být kladné číslo menší než 1 a udává počet platných desetinných míst výsledku.

$$\sin(x) = \frac{x^1}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, \quad |x| < +\infty$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots, \quad |x| < +\infty$$

$$e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, \quad |x| < +\infty$$

$$\ln(x) = 2 \left(\frac{(x-1)}{(x+1)} + \frac{(x-1)^3}{3 \cdot (x+1)^3} + \frac{(x-1)^5}{5 \cdot (x+1)^5} + \dots \right), \quad x > 0$$

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right)$$

10. Třídu `MathTools` doplňte o metodu pro rekurentní výpočet odmocniny zadaného čísla, výpočet proveďte se zadanou přesností.

Zadané parametry výpočtu: x , `eps`

První odhad: $y_1 = x$

Další aproximace: $y_{i+1} = \frac{1}{2} \left(\frac{x}{y_i} + y_i \right)$, Newtonův iterační vztah

Použití zadané přesnosti: výpočet ukončit v případě, že absolutní hodnota rozdílu dvou po sobě aproximovaných hodnot je menší než požadovaná přesnost: $|y_{i+1} - y_i| < eps$.

Při testování porovnejte výsledek vlastní metody s výsledkem metody `Math.sqrt()`

Dále implementujte stejnou výpočet stejné hodnoty (druhé odmocniny) s „absolutní přesností“ (přesnost typu `double`).