# **Dokumentace – úkol 03**

Stanislav Čech, Ondřej Kohout, Kryštof Malinda, Petr Svoboda

## **První část**

### Odmocnina

Testovací třída OdmocninaTest ověřuje funkčnost metody Math.sqrt() z knihovny Java. Tato metoda slouží k výpočtu druhé odmocniny z reálného čísla. Test je napsán pomocí JUnit 5 a je umístěn v balíčku test.

V testovací metodě testOdmocnina() jsou otestovány tři základní vstupy: 9.0, 2.0 a 0.0. Pro každý z nich je ověřeno, zda metoda vrací správnou odmocninu. Dále jsou testovány i okrajové případy: odmocnina ze záporného čísla, která má vrátit NaN, a odmocnina z Double.POSITIVE\_INFINITY, kde se očekává výsledek nekonečno.

Testy slouží jako výukový příklad použití JUnit pro ověřování výstupu standardních knihovních metod a pokrývají nejen běžné vstupy, ale i extrémní a neočekávané hodnoty. Pomáhají pochopit chování matematických funkcí v jazyce Java a zároveň ukazují dobré praktiky při psaní testů.

### Goniometrické funkce

Třída GoniometrickeFunkceTest testuje správnost výpočtu goniometrických funkcí sinus a tangens pro úhly zadané ve stupních. K výpočtu je využita standardní knihovna Math v jazyce Java, která pracuje s úhly v radiánech, proto je nutná konverze pomocí Math.toRadians().

V testovací metodě testGoniometrickeFunkce() jsou použity tři úhly: 30°, 45° a 90°. Pro každý úhel je spočítán sinus, pro první dva také tangens. Výsledky jsou porovnávány s očekávanými hodnotami pomocí metody assertEquals s malou tolerancí kvůli desetinným zaokrouhlením.

Tangens 90° není v matematice definován, protože se blíží nekonečnu. V testu je proto ověřeno, že jeho hodnota je extrémně vysoká, což odpovídá správnému chování funkce Math.tan() pro tento vstup.

Test slouží jako výukový příklad toho, jak pracovat s úhly ve stupních v prostředí Javy, jak provádět konverzi do radiánů a jak ověřovat výsledky pomocí JUnit testů.

### List

Třída ListIndexOfTest testuje metodu indexOf() z rozhraní List v jazyce Java. Tato metoda vrací index prvního výskytu daného prvku v seznamu nebo -1, pokud prvek v seznamu neexistuje. Test využívá implementaci ArrayList.

V první části testu testListIndexOf() je vytvořen seznam řetězců, do kterého se přidávají hodnoty jako "jablko", "hruška" a "švestka". Test ověřuje, zda metoda indexOf() správně vrací indexy těchto prvků. Zároveň kontroluje chování metody při hledání hodnoty, která v seznamu neexistuje — v tomto případě "banán", kde se očekává výsledek -1.

Ve druhé části testu se pracuje se seznamem objektů. Testuje se, že indexOf() porovnává prvky pomocí referencí (nikoli pouze podle hodnoty). Tím je ověřeno, že nový objekt, byť stejného typu, nebude považován za již existující v seznamu, pokud se nejedná o tu samou instanci.

Tento test demonstruje důležité vlastnosti metody indexOf() — jak funguje s hodnotovými typy (např. String s přepsanou metodou equals) i s objekty, kde záleží na referenci. Slouží jako výukový příklad pro práci s kolekcemi v Javě.

### Slovník

Třída HashMapTest testuje chování třídy HashMap, která v Javě slouží jako implementace slovníku (asociativního pole). Zaměřuje se především na metodu put(), která do mapy vkládá hodnoty a vrací předchozí hodnotu daného klíče, pokud existovala.

V první části testu se ověřuje, že při přidání nových klíčů metoda put() vrací null a následné volání get() vrací správně vložené hodnoty. Poté je testováno přepsání existující hodnoty — pro klíč "jablko" je nastavena nová hodnota a zkontroluje se nejen nová hodnota, ale i to, že put() vrátil starou hodnotu.

Dále test ověřuje, že HashMap v Javě umožňuje vložit null jako klíč i jako hodnotu. Toto je důležitá vlastnost, protože ne všechny datové struktury to podporují.

Na závěr se kontroluje celkový počet položek ve slovníku, přítomnost a nepřítomnost konkrétních klíčů pomocí metod size(), containsKey() a get().

Test slouží jako přehledný výukový příklad použití HashMap, její metody put() a práce s hodnotami a klíči včetně null.

## **Druhá část**

Třída ElektrickyDvojbran reprezentuje elektrický dvojbran pomocí přenosových parametrů A, B, C a D, které je možné zadat ručně, nebo vytvořit z přednastavených hodnot jako například napěťový dělič přes metodu vytvorNapetovyDelic. Hlavní metodou je vypocitejVystupy, která na základě vstupního napětí a zátěže vypočítá výstupní napětí a proud. Program také umožňuje zobrazit mezivýpočty (zobrazMezivypocty) i výsledky výpočtu (zobrazVysledky) a obsahuje statickou metodu main, která umožňuje uživateli zadávat hodnoty z příkazové řádky a zobrazit odpovědi. Pro ověření správnosti chování třídy slouží testovací třída ElektrickyDvojbranTest využívající JUnit 5. Obsahuje testy pro běžné i mezní případy, ověřuje funkčnost napěťového děliče, porovnává výstupy obecného a zjednodušeného výpočtu a kontroluje správnost výsledků při různých parametrech a zátěži. Testy potvrzují korektní implementaci metod a celkovou spolehlivost výpočtového modelu.

## **Třetí část**

### Calculator

Třída Calculator představuje jednoduchou grafickou kalkulačku v Javě s použitím knihovny Swing. Uživatel zadá dvě čísla a zvolí operaci (+, –, \*, /) z rozbalovací nabídky, přičemž po kliknutí na tlačítko „Vypočítat“ se zobrazí výsledek ve výstupním poli. Kalkulačka kontroluje platnost vstupů a ošetřuje dělení nulou pomocí dialogového okna s chybovou hláškou. Tlačítko „Vymazat“ slouží k resetování všech polí. Rozhraní je přehledně rozloženo pomocí GridLayout a program se spouští v metodě main, kde se inicializuje a zobrazí hlavní okno.

### Heslo

Třída PasswordDialog vytváří grafické rozhraní pro zadání a potvrzení hesla pomocí Swingu. Uživatel zadává heslo a jeho potvrzení, přičemž viditelnost lze přepínat pomocí checkboxu. Aplikace v reálném čase vyhodnocuje sílu hesla a zobrazuje ji pomocí popisu a barevného progress baru. Síla hesla se určuje podle délky, různorodosti znaků a přítomnosti speciálních znaků. Tlačítko „Potvrdit“ je aktivní pouze tehdy, když se obě pole shodují a síla hesla dosáhne minimální hodnoty. Po úspěšném zadání se zobrazuje potvrzení a formulář se resetuje.