

Programovací jazyk Java

CVIČENÍ 6

Úkoly

Pokračujte ve třídě **FileOperations** z minulého cvičení, případně si založte nový projekt a v něm novou třídu **FileOperations**.

Vytvořte statickou metodu **long getTotalSizeOfFilesByExtension(File dir, String ext)**. Metoda očekává existující adresář jako argument typu **File** a souborovou příponu (např. "pdf"). Pokud při volání metody bude některý z argumentů zadána hodnota **null**, nebo jako **dir** bude zadán neexistující adresář, nebo bude zadán soubor a ne adresář, nebo nebude možné z adresáře číst, metoda vyhodí výjimku **IllegalArgumentException** s odpovídajícím chybovým hlášením.

Metoda v adresáři **dir** vyhledá všechny soubory se zadanou příponou **ext** a vrací celkovou velikost těchto souborů (v bajtech). Metoda prohledává celou podadresářovou větev.

Dále do třídy **FileOperations** přidejte následující metodu:

```
public static File chooseDirectory() {
    JFileChooser chooser = new JFileChooser();
    chooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    chooser.setDialogTitle("Vyberte složku");

    File selectedDir = null;
    while (selectedDir == null) {
        int result = chooser.showOpenDialog(null);
        if (result == JFileChooser.APPROVE_OPTION) {
            selectedDir = chooser.getSelectedFile();
        } else {
            JOptionPane.showMessageDialog(null,
                "Musíte vybrat složku, jinak nelze pokračovat.",
                "Výběr složky",
                JOptionPane.WARNING_MESSAGE);
        }
    }
    return selectedDir;
}
```

Dále do třídy **Main** přidejte následující metody:

```
public static String askForText(String message) {
    return JOptionPane.showInputDialog(null, message, "Zadejte text",
        JOptionPane.PLAIN_MESSAGE);
}

public static void showResult(long size) {
    JOptionPane.showMessageDialog(null,
```

```

        "Celková velikost souborů: " + size + " bajtů.",
        "Výsledek",
        JOptionPane.INFORMATION_MESSAGE);
    }
}

```

Ve třídě **Main** v metodě main si pomocí metody **askForText** vyžádejte od uživatele příponu souborů. Dále si pomocí metody **chooseDirectory()** třídy **FileOperations** požádejte uživatele o výběr složky. Následně vybranou složku a zadanou příponu zadejte metodě **getTotalSizeOfFilesByExtension** třídy **FileOperations**. Pokud metoda **getTotalSizeOfFilesByExtension** vyhodí výjimku, volejte vyzvěte znova uživatele k výběru složky a znova zavolejte metodu **getTotalSizeOfFilesByExtension**. Jakmile tato metoda úspěšně skoční (bez vyvolání výjimky), předejte získanou hodnotu celkové velikosti souborů metodě **showResult** k zobrazení uživateli.

Vytvořte třídu **Osoba** s proměnnými **jmeno** (String), **muz** (boolean) a **vyska** (int). Bezparametrický konstruktor nastaví hodnoty proměnných následovně:

- **muz** → náhodná hodnota typu boolean,
- **vyska** → náhodné celé číslo z intervalu 150–200,
- **jmeno** → sekvence 2–5 náhodných malých písmen angl. abecedy.

(Pro generování náhodných hodnot můžete využít metody instance třídy **java.util.Random**.)

Překryjte metodu **toString** tak, aby vracela hodnoty všech proměnných.

Překryjte metodu **equals** tak, aby platilo:

- Pokud je porovnáván objekt sám se sebou, metoda vrací **true**.
- Pokud objekt není instancí třídy Osoba (nebo její podtřídy), metoda vrací **false**.
- Pokud jsou si rovny hodnoty všech proměnných, metoda vrací **true**, jinak **false**.

Případnou metodu **hashCode()** z třídy **Osoba** odstraňte.

Ve stejném souboru vytvořte následující podtřídy třídy Osoba s požadovanou implementací metody **hashCode()**:

- Třída **ExtraNehodnaOsoba** – metoda **hashCode()** vrací hodnotu 5.
- Třída **NehodnaOsoba** – metoda **hashCode()** vrací hodnotu proměnné **vyska**.
- Třída **OsobaBezHashCode** – metoda **hashCode()** není implementována (tj. dědí se implementace z třídy **Object**).
- Třída **PrijatelnaOsoba** – metoda **hashCode()** vrací součin hodnoty proměnné **vyska** a počtu znaků proměnné **jmeno**.
- Třída **PerfektniOsoba** – metoda **hashCode()** je implementována následovně:

```

@Override
public int hashCode() {
    int vysledek = this.muz ? 1 : 0;
    vysledek = 31 * vysledek + vyska;
}

```

```
    vysledek = 31 * vysledek + jmeno.hashCode();
    return vysledek;
}
```

- Třída **ObjectsHashOsoba** – metoda **hashCode()** vrací výsledek volání statické metody **hash** třídy **Objects**, které předá hodnoty všem proměnných objektu.
- Třída **ObjectsHashCodeOsoba** – metoda **hashCode()** vrací výsledek volání statické metody **hashCode** třídy **Objects**, které předá odkaz na vlastní objekt.

Vytvořte proměnnou typu pole se 100 000 instancemi třídy **ExtraNevhodnaOsoba**. Dále vytvořte neseřazenou množinu typu **HashSet** s výchozí kapacitou 100 000. V cyklu do množiny vložte všechny prvky pole osob. Změřte, kolik času tato operace zabere, využijte **System.currentTimeMillis()**.

Dále pomocí metody **contains** ověřte, zda jsou v množině všechny prvky z pole osob. Opět změřte, kolik času tato operace zabere. Porovnejte výsledky pro pole osob různých typů (**ExtraNevhodnaOsoba**, **NevhodnaOsoba**, ...).