Pole v Javě

Java obsahuje dva neprimitivní datové typy a to pole a objekty. Proměnné těchto typů jsou označovány jako **referenční**.

🛈 Pole jsou indexována od [0] do [jmenoPole.length - 1]

🛈 Indexy pole mohou být pouze konstanty, proměnné nebo výrazy typu **int**

Deklarace Pole

Skládá se ze dvou částí - z typu pole a z jeho jména. Při deklaraci se neudává velikost pole, protože všechna pole jsou alokována dynamicky, takže velikost pole určuje až při žádosti o jeho vytvoření.

int[] pole;

Pouhou deklarací pole nepřidělíme proměnné žádnou paměť, proto je nutné před prvním použitím pole tuto paměť přidělit pomocí operátoru **new**, např.:

pole = new int[];

🛈 Kompilátor zaručuje, že prvky takto vytvořeného pole budou mít **nulové hodnoty**. Pro pole typu boolean, pak hodnoty **false**, a jsou-li prvky referenční proměnné, pak **null**.

Délka pole

Délku pole zadanou při jeho vytvoření si **nemusíme pamatovat**, protože ji zjistíme pomocí atributu se jménem length. Tuto konstantu vlastní automaticky jakékoliv pole.

System.out.println(“Delka pole: “ + pole.length);

🛈 Narozdíl od jazyka C je za běhu programu striktně kontrolované překročení obou mezí pole. Program pak reaguje vyhozením vyjímky ArrayOutOfBoundsException.

Inicializace pole

Pole nemusíme vytvářet pouze pomocí operátoru new. Můžeme ho rovnou při deklaraci naplnit hodnotami pomocí statického inicializátoru, např.:

int[] prvocisla = { 1, 2, 3, 5, 7, 11 };

Dvourozměrná pole

Je to pole polí a pracuje se s nimi obdobně jako s jednorozměrnými poli, jen je nutné při deklaraci zadat všechny rozměry.

int[][] polepoli = new int[5][4];

// 5 -> pocet radku, 4 -> pocet sloupcu

polepoli.length // pocet radku

polepoli[0].length // pocet sloupcu

ArrayList

Hlavním rozdílem mezi ArrayListem a základním polem v Javě je, že pole po jeho vytvoření nemůže měnit počet prvků, lze ho pouze vytvořit znovu. Do ArrayListu můžeme přidávat nové položky nebo je odebírat.

Jak ArrayList, tak pole může obsahovat položky pouze jednoho datového typu.

Vytvoření ArrayListu

Pro jeho vytvoření je nutné uvést v ostrých závorkách datový typ, který bude ArrayList obsahovat. A opět pomocí operátoru new mu přiřadit paměť.

ArrayList<Integer> listIntu = new ArrayList<>();

🛈 Pro vytvoření arraylistu s položkami základního datového typu musíme použít jejich objektovou reprezentaci (Integer, Character, Boolean, Double… )

🛈 Při přiřazování paměti již není nutné uvádět datový typ v ostrých závorkách, Java si jej umí zjistit sama, podle proměnné, do které paměť přiřazujeme

Základní metody pro práci s ArrayListem

listIntu.add(1); // pridani cisla 1 do pole

listIntu.set(0, 3); // zmena cisla na indexu 0 cislem 3

listIntu.remove(0); // odebrani polozky s indexem 0

listIntu.clear(); // vymazani celeho listu

listIntu.size(); // vrati pocet polozek v listu

🛈 Pokud list obsahuje položky referenčního datového typu, můžeme do metody remove() vložit odkaz na danou polozku, čímž ji odstraníme bez toho, abychom věděli index této položky