# 11. Standardní datové typy jazyka JAVA (rozdělení, rozsahy)

## Datový Typ

V programování definuje druh nebo význam hodnot, kterých smí nabývat proměnná (konstanta). Datový typ je určen oborem hodnot a zároveň výpočetními operacemi, které lze s hodnotami tohoto typu provádět

## Proměnné

Místo v paměti dle typu, který říká, jakou bude mít velikost a jaké operace s proměnou se dají provádět. Vytvoření proměnné se skládá z názvu typu a jména proměnné. V proměnných se můžou uchovávat celá čísla, znaky, nebo ukazatele na objekt. Proměnná na rozdíl od konstant umožňuje měnit svojí hodnotu.

Deklarace proměnné je vytvoření místa v paměti dle určitého typu. Následně je doporučená inicializace (přiřazení) počáteční hodnoty proměnné, není nutná, ale doporučuje se provádět vždy, kdy je to možné.

## Konstanty

Konstanty v jazyce Java jsou stejné jako proměnné s tím rozdílem že se před název typu napíše klíčové slovo final. Konstanta nemůže během svého života měnit svou hodnotu.

Deklarace konstanty je stejně jako u proměnných vytvoření místa v paměti určitého typu. Pokud je známo, jaká hodnota se bude do konstanty přiřazovat, pak se může inicializovat ihned. Jinak se konstanta nechá neinicializovaná, a hodnota se jí přiřadí, až bude známa.

## Pojmenování

Názvy proměnných a konstant by měli odpovídat jejich použití.

Názvy proměnných musí začínat malým písmenem, dále pokračují také malými písmeny, pokud se název skládá z několika slov, použije se velbloudíNotace (camelCase).

Názvy konstant se udávají velkými písmeny. Pokud se skládá z více slov, používá se podtržítko.

#### public static final double KONSTANTA\_PI = 3.14159265359;

## Primitivní datové typy

Java má 8 primitivních datových typů. Procesor s primitivními datovými typy může pracovat přímo 🡪 rychlá práce s těmito typy. Každý primitivní datový typ může nabývat různých hodnot a hodí se pro jinou práci. Jedná se o čtyři celočíselné typy (byte, short, int, long), dva typy pro práci s plovoucí desetinou čárkou (float, double), znakový typ (char), který nabývá hodnot z ASCII tabulky. A poslední je pravdivostní typ (boolean).

## Celočíselné primitivní datové typy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primitivní typ** | **Velikost** | **Minimum** | **Maximum** | **Wrapper Class** |
| Byte | 1B | -27 | -27 -1 | Byte |
| Short | 2B | -215 | -215 -1 | Short |
| Int | 4B | -231 | -231 -1 | Integer |
| Long | 8B | -263 | -263 -1 | Long |

## Reálné primitivní datové typy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primitivní typ** | **Velikost** | **Minimum** | **Maximum** | **Wrapper Class** |
| Float | 4B | ANSI IEEE.754 | ANSI IEEE.754 | Float |
| Double | 8B | ANSI IEEE.754 | ANSI IEEE.754 | Double |

## Pravdivostní primitivní datový typ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primitivní typ** | **Velikost** | **Minimum** | **Maximum** | **Wrapper Class** |
| boolean | Není přesně definována | false | true | Boolean |

## Znakový primitivní datový typ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primitivní typ** | **Velikost** | **Minimum** | **Maximum** | **Wrapper Class** |
| Char | 2B | Unicode 0 | Unicode 216-1 | Character |

#### char znak = 65; // ASCCI hodnota písmene A

#### char znak = `A`;

#### char znak = `u00ff`;

## Datový typ void

Nelze vytvořit proměnnou datového typu void. Datový typ void pouze udává, že metoda nemá návratovou hodnotu.

## Wrapper Class

Obalová třída, každý z 8 primitivních datových typů má „Wrapper Class“, která určuje, co lze provádět s daným datovým typem.

## Objektové datové typy

„Referenční datové typy“. Jsou datové typy objektu. Hodnota referenční proměnné (proměnná s objektem) je odkaz na místo v paměti, kde je daný objekt (pole) uložen. Deklarace je téměř totožná s vytvořením normální proměnné, jen místo názvu se uvádí název třídy objektu.

#### String retezec = “Cosi”;

#### public class cosi {

#### public static void main(String[] args) {

#### float a = 5;

#### float b = 0;

#### System.*out*.println(a / b); //Infinity

#### float aa = 0;

#### float bb = 0;

#### System.*out*.println(aa / bb); //Not a Number (NaN)

#### }

#### }