

Hornerovo schéma

komentář k ukázkovému řešení

V úloze máme za úkol nejprve načíst mnohočlen s celočíselnými koeficienty, pak jej vypsát ve dvou jiných formátech a nakonec vypsát jeho hodnoty v zadaných bodech.

Víme, že maximální stupeň zadaného mnohočlenu bude 100, takže nám na reprezentaci mnohočlenu v paměti postačí jediné celočíselné pole o 101 prvcích, v němž budou uloženy koeficienty polynomu. Jednotlivé exponenty nikam ukládat nemusíme, protože ty budou ve skutečnosti představovány indexy pole. Například hodnota 311 uložená v prvku s indexem 5 znamená, že mnohočlen obsahuje člen $311 \cdot x^5$, apod., To, že některé prvky pole koeficientů zůstanou nevyužité, nás při této paměťové zátěži — 404 bytů — jistě nemusí trápit. Mnohočlen, který je zadán na vstupu, může obsahovat více členů se stejným exponentem, pro tento složitější tvar si však žádnou vnitřní reprezentaci chystat nebudeme, protože s tímto tvarem v programu dále již pracovat nebudeme. Pokusíme se (asi úspěšně) odfiltrvat tuto složitost hned při načítání mnohočlenu do paměti.

Budeme pouze potřebovat navíc ještě jednu proměnnou, v níž bude uložen stupeň polynomu, což je nejvyšší hodnota exponentu ze všech nenulových členů mnohočlenu.

Načtení si můžeme zjednodušit tak, že jedním průchodem vstupního řetězce přeneseme všechny znaky kromě mezer do pomocného pole. Poté opět jednou projdeme pomocné pole znaků, extrahujeme z něj postupně všechny koeficienty a exponenty a během procházení budeme průběžně aktualizovat hodnoty pole koeficientů. Koncepčně stačí, abychom dokázali přečíst jediný člen mnohočlenu, opakováním této akce v cyklu a přičítáním(!) nalezených koeficientů do příslušných prvků pole koeficientů podle nalezených exponentů zkonstruujeme snadno celou reprezentaci mnohočlenu. Přečtení jednoho členu je ale jednoduché, víme, že nejprve musíme přečíst znaménko, pak koeficient, pak znaménko násobení, pak znak proměnné, znak umocňování a exponent. Některé tyto prvky mohou v členu chybět, ale pravidla pro jejich přítomnost/chybění jsou jednoduchá, odkazujeme čtenáře na příložený kód, který by měl být výmluvnější než dlouhé popisy.

Jakmile mnohočlen načteme a získáme jeho interní reprezentaci, je zbytek programu snadný. Vypsát mnohočlen znamená vypsát každý člen zvlášť. Vypsát jeden člen s případně chybějícím znaménkem plus, pokud jde o nejvyšší exponent, je akce podobně přímočará jako načtení členu, opět odkazujeme čtenáře na příložený kód.

Výpis v podobě Hornerova schématu lze provést krátkou rekurzivní funkcí, nebo funkcí nepatrně delší bez rekurze. Stejně tak Hornerovo schéma použijeme při výpočtu hodnot polynomu v daných bodech, protože právě pro tuto činnost bylo původně navrženo.

Poznámka k časové a paměťové efektivitě

Zjednodušené načítání s úvodním odstraněním mezer jsme volili jen pro přehlednější a kompaktnější kód. Pokud bychom načítali mnohočlen přímo ze vstupního řetězce obsahujícího všechny mezery, nepotřebovali bychom pomocné pole znaků, stačilo by pouze vložit volání funkce přeskakující mezery před každý posun ukazatele polohy ve vstupním řetězci. Méně sběhlým čtenářům to doporučujeme provést jako jednoduché cvičení.

Technická poznámka k pomocnému poli znaků

Nepoučená manipulace s typem String v jazyce Java snadno vede při větším množství dat ke katastrofálnímu zpomalení programu díky časově velmi neefektivní koncepci této třídy. Někteří řešitelé se o tom nepříjemně přesvědčili na vlastní kůži. V úlohách podobného rázu je proto přinejmenším vhodné, zejména v Javě, používat co nejjednodušší datové typy, kde má programátor lepší představu, co ve skutečnosti program s jeho kódem dělá. Z tohoto důvodu je v ukázkovém řešení použito právě jen pole znaků.