

Kód na standardním vstupu v jazyce IPPcode24 má v prvním kroku odstraněny všechny prázdné řádky a ty řádky, které obsahují pouze komentář. Následně je provedeno rozdělení na jednotlivé řádky, které budou postupně převáděny do XML. Převáděný kód se ukládá do bufferu a je vytisknut na standardní výstup pouze v případě, kdy je původní kód bezchybný.

První řádek musí obsahovat korektně napsanou hlavičku. Krom této jediné výjimky se na ostatních řádcích budou vyskytovat jednotlivé instrukce a vše s nimi spojeno. Mezi instrukcí a jejími operandy musí existovat oddělení ve formě mezery či tabulátorů. Když je tedy procházen nějaký řádek původního kódu, je rozdělen na jednotlivé slova nebo úseky znaků, které byly odděleny od sebe.

Slovo na první pozici je vždy instrukce z instrukční sady. Pro účely převodu do XML nás nezajímá dělení instrukcí tak, jak je v dokumentaci IPPcode24, resp. dle jejich účelu, nýbrž potřebujeme vědět, kolik a jakých operandů bude po přečtené instrukci následovat. Pokračuje se v převádění řádku a konec nastává buď v případě převodu posledního operandu, nalezení začátku komentáře či existence lexikální či syntaktické chyby zdrojového kódu. Až při správném zakončení převodu řádku dojde ke kontrole celkového počtu operandů dle požadavků konkrétní instrukce.

Dochází k určení datového typu dle obsahu operandu před prvním výskytem zavináče. V případě neexistence zavináče se jedná o návěští. Krom těchto návěští nastává kontrola druhé části operandu (po výskytu zavináče), kde každý datový typ má specifické potřeby a vede to na jinou kontrolní rutinu. Nejenže musí být celý operand korektně zapsán, ale ještě se zkontroluje, zda datový typ sedí dle potřeb k instrukci na konkrétním řádku.

Ke kontrole druhé části operandu se využívají primárně regulární výrazy. Jsou navrženy dle specifikací v dokumentaci IPPcode24 a příslušného datového typu. Příkladem může být následující výraz pro kontrolu správnosti proměnné:

$$^{[a-zA-Z_\\-!\\%\\$\\&*?]} [a-zA-Z0-9_\\-!\\%\\$\\&*?]*$$$

U proměnných a řetězců dochází navíc v tomto kroku k převodu problematických znaků v XML a jsou nahrazené odpovídajícími XML entitami.

Dojde-li k bezchybnému překladu posledního řádku, vypíše se obsah bufferu na standardní výstup a program končí.