

# Konkrétní datové typy – opakování a rozšíření

## Programovací techniky

doc. Ing. Jiří Rybička Dr.  
ústav informatiky  
PEF MENDELU v Brně  
`rybicka@mendelu.cz`

Datový typ

Jednoduché typy

Strukturované typy

- Specifikace: množina povolených hodnot a množina povolených operací

- Specifikace: množina povolených hodnot a množina povolených operací
- Konkrétní datový typ: typ implementovaný v daném jazyce

- Specifikace: množina povolených hodnot a množina povolených operací
- Konkrétní datový typ: typ implementovaný v daném jazyce
- Každý jazyk má jiný repertoár konkrétních typů

- Specifikace: množina povolených hodnot a množina povolených operací
- Konkrétní datový typ: typ implementovaný v daném jazyce
- Každý jazyk má jiný repertoár konkrétních typů
- Datové typy pro stejný účel jsou v různých jazycích různě implementovány

- Specifikace: množina povolených hodnot a množina povolených operací
- Konkrétní datový typ: typ implementovaný v daném jazyce
- Každý jazyk má jiný repertoár konkrétních typů
- Datové typy pro stejný účel jsou v různých jazycích různě implementovány
- Jazyk Pascal: Datové typy jednoduché, strukturované, dynamické, objektové.

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy



Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- V jednom okamžiku uchovávají jednu hodnotu

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- V jednom okamžiku uchovávají jednu hodnotu
- Základní členění: ordinální, neordinální

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- V jednom okamžiku uchovávají jednu hodnotu
- Základní členění: ordinální, neordinální
- Ordinální typy: jejich hodnoty jsou zobrazeny na vhodné podmnožinu celých čísel (implementace)

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- V jednom okamžiku uchovávají jednu hodnotu
- Základní členění: ordinální, neordinální
- Ordinální typy: jejich hodnoty jsou zobrazeny na vhodné podmnožině celých čísel (implementace)
- Neordinální typy: racionální čísla (zobrazeny pomocí dvojic celých čísel – mantisa, exponent)

- V jednom okamžiku uchovávají jednu hodnotu
- Základní členění: ordinální, neordinální
- Ordinální typy: jejich hodnoty jsou zobrazeny na vhodnou podmnožinu celých čísel (implementace)
- Neordinální typy: racionální čísla (zobrazeny pomocí dvojic celých čísel – mantisa, exponent)
- Ordinální typy: celá čísla (integer – základní ve všech verzích, rozšiřující jsou shortint, smallint, longint, byte, word, longword, cardinal); boolean, char, interval, výčet.

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Operace vyplývají ze zobrazení do množiny celých čísel

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Operace vyplývají ze zobrazení do množiny celých čísel
- Základní: succ(X), pred(X), porovnání (6 operátorů)



Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Operace vyplývají ze zobrazení do množiny celých čísel
- Základní: succ(X), pred(X), porovnání (6 operátorů)
- Konverzní: ord(X), typ(I)

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Operace vyplývají ze zobrazení do množiny celých čísel
- Základní: succ(X), pred(X), porovnání (6 operátorů)
- Konverzní: ord(X), typ(I)
- Rozšíření: Inc(X, [K]), Dec(X, [K])

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód
- Logické vlastnosti: každý bit jako jedna logická hodnota

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslelných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód
- Logické vlastnosti: každý bit jako jedna logická hodnota
- Rozšíření operátorů **and**, **or**, **not**.

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet, Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód
- Logické vlastnosti: každý bit jako jedna logická hodnota
- Rozšíření operátorů **and**, **or**, **not**.
- Příklady:  
5 **or** 9 = 13    **not** 255 = 0  
5 **and** 9 = 1    5 **xor** 9 = 12

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslelných typech

Využití bitových operací

Znakový typ

Výčet, Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód
- Logické vlastnosti: každý bit jako jedna logická hodnota
- Rozšíření operátorů **and**, **or**, **not**.
- Příklady:  
     $5 \text{ or } 9 = 13$     **not** 255 = 0  
     $5 \text{ and } 9 = 1$      $5 \text{ xor } 9 = 12$
- Bitové posuvy: **shr**, **shl**



Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód
- Logické vlastnosti: každý bit jako jedna logická hodnota
- Rozšíření operátorů **and**, **or**, **not**.
- Příklady:  
     $5 \text{ or } 9 = 13$      $\text{not } 255 = 0$   
     $5 \text{ and } 9 = 1$      $5 \text{ xor } 9 = 12$
- Bitové posuvy: **shr**, **shl**
- Posuv vlevo – do nejnižšího bitu se vloží vždy nula

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Celá čísla: aritmetické vlastnosti, doplňkový kód
- Logické vlastnosti: každý bit jako jedna logická hodnota
- Rozšíření operátorů **and**, **or**, **not**.
- Příklady:  
 $5 \text{ or } 9 = 13$      $\text{not } 255 = 0$   
 $5 \text{ and } 9 = 1$      $5 \text{ xor } 9 = 12$
- Bitové posuvy: **shr**, **shl**
- Posuv vlevo – do nejnižšího bitu se vloží vždy nula
- Posuv vpravo – dva typy:
  - a) do nejvyššího bitu se vloží nula = **logický posuv**
  - b) do nejvyššího bitu se zopakuje původní hodnota = **aritmetický posuv**

# Využití bitových operací

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Umístění více logických informací do minimálního prostoru  
1001 = archivní, ne systémový, ne read only, adresář

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Umístění více logických informací do minimálního prostoru  
1001 = archivní, ne systémový, ne read only, adresář
- Maskování  
x `and` 3 – výběr nejnižších dvou bitů

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet, Interval

Strukturované typy

- Umístění více logických informací do minimálního prostoru  
1001 = archivní, ne systémový, ne read only, adresář
- Maskování  
X `and` 3 – výběr nejnižších dvou bitů
- Zjištění hodnot vybraných bitů – příklady: zabezpečení paritou, šifrování  
kombinace `odd(X)` a posuvu vpravo

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy

- Umístění více logických informací do minimálního prostoru  
1001 = archivní, ne systémový, ne read only, adresář
- Maskování  
 $x \text{ and } 3$  – výběr nejnižších dvou bitů
- Zjištění hodnot vybraných bitů – příklady: zabezpečení paritou, šifrování  
kombinace  $\text{odd}(X)$  a posuvu vpravo
- Násobení a dělení hodnotou se základem 2  
násobení nahrazeno posuvem doleva, dělení posuvem doprava

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

Strukturované typy



## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet. Interval

### Strukturované typy

- Znakové hodnoty v jednobytovém kódování

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

### Strukturované typy

- Znakové hodnoty v jednobytovém kódování
- Zápis hodnot pomocí obrazů v apostrofech nebo pomocí ordinálního čísla, např. #10

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

### Strukturované typy

- Znakové hodnoty v jednobytovém kódování
- Zápis hodnot pomocí obrazů v apostrofech nebo pomocí ordinálního čísla, např. #10
- Operace: ordinální + UpCase

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

### Strukturované typy

- Znakové hodnoty v jednobytovém kódování
- Zápis hodnot pomocí obrazů v apostrofech nebo pomocí ordinálního čísla, např. #10
- Operace: ordinální + UpCase
- Historická funkce `chr`

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet. Interval

### Strukturované typy

- Znakové hodnoty v jednobytovém kódování
- Zápis hodnot pomocí obrazů v apostrofech nebo pomocí ordinálního čísla, např. #10
- Operace: ordinální + UpCase
- Historická funkce `chr`
- Znakový kód ASCII:

řídící znaky	000x xxxx,
mezera	0010 0000
čísllice	0011 xxxx,
velká písmena	010x xxxx,
malá písmena	011x xxxx,
národní znaky	1xxx xxxx

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet, Interval

Strukturované typy

Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet, Interval

Strukturované typy

- Výčet: Hodnotami jsou identifikátory

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet, Interval

### Strukturované typy

- Výčet: Hodnotami jsou identifikátory
- Implementace celými čísly počínaje nulou



Datový typ

Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet, interval

Strukturované typy

- Výčet: Hodnotami jsou identifikátory
- Implementace celými čísly počínaje nulou
- Identifikátory představují pojmenované konstanty

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíselných typech

Využití bitových operací

Znakový typ

Výčet, Interval

### Strukturované typy

- Výčet: Hodnotami jsou identifikátory
- Implementace celými čísly počínaje nulou
- Identifikátory představují pojmenované konstanty
- Hlavní výhodou je zvýšení čitelnosti zdrojového textu

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet, Interval

### Strukturované typy

- Výčet: Hodnotami jsou identifikátory
- Implementace celými čísly počínaje nulou
- Identifikátory představují pojmenované konstanty
- Hlavní výhodou je zvýšení čitelnosti zdrojového textu
- Nutnost uživatelských konverzí při textovém vstupu a výstupu

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na  
celočíslných typech

Využití bitových operací

Znakový typ

Výčet, Interval

### Strukturované typy

- Výčet: Hodnotami jsou identifikátory
- Implementace celými čísly počínaje nulou
- Identifikátory představují pojmenované konstanty
- Hlavní výhodou je zvýšení čitelnosti zdrojového textu
- Nutnost uživatelských konverzí při textovém vstupu a výstupu
- Interval – zúžení hodnot a operací na podmnožinu jiného ordinálního typu

## Datový typ

### Jednoduché typy

Ordinální typy

Bitové operace na celočíselných typech

Využití bitových operací

Znakový typ

Výčet, Interval

### Strukturované typy

- Výčet: Hodnotami jsou identifikátory
- Implementace celými čísly počínaje nulou
- Identifikátory představují pojmenované konstanty
- Hlavní výhodou je zvýšení čitelnosti zdrojového textu
- Nutnost uživatelských konverzí při textovém vstupu a výstupu
- Interval – zúžení hodnot a operací na podmnožinu jiného ordinálního typu
- Určen ke kontrole rozsahů hodnot, při zapnutí kontrol překladače

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- V jednom okamžiku uchovávají více hodnot

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- V jednom okamžiku uchovávají více hodnot
- Lze rozdělit na homogenní (složky stejného typu) a heterogenní (složky různých typů)



Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- V jednom okamžiku uchovávají více hodnot
- Lze rozdělit na homogenní (složky stejného typu) a heterogenní (složky různých typů)
- **Bázový typ** = typ složky

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- V jednom okamžiku uchovávají více hodnot
- Lze rozdělit na homogenní (složky stejného typu) a heterogenní (složky různých typů)
- **Bázový typ** = typ složky
- Základní operací je přístup ke složce

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- V jednom okamžiku uchovávají více hodnot
- Lze rozdělit na homogenní (složky stejného typu) a heterogenní (složky různých typů)
- **Bázový typ** = typ složky
- Základní operací je přístup ke složce
- Paměťové typy lze přiřazovat

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- V jednom okamžiku uchovávají více hodnot
- Lze rozdělit na homogenní (složky stejného typu) a heterogenní (složky různých typů)
- **Bázový typ** = typ složky
- Základní operací je přístup ke složce
- Paměťové typy lze přiřazovat
- Specifické možnosti: řetězec, množina, soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Pole představuje nejbližší obraz operační paměti

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Pole představuje nejbližší obraz operační paměti
- Koncepce pole v jazyce Pascal – vždy statické, jednorozměrné

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Pole představuje nejbližší obraz operační paměti
- Koncepce pole v jazyce Pascal – vždy statické, jednorozměrné
- Složkou pole může být libovolný typ bez omezení



Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Pole představuje nejbližší obraz operační paměti
- Koncepce pole v jazyce Pascal – vždy statické, jednorozměrné
- Složkou pole může být libovolný typ bez omezení
- Přístup ke složce pomocí indexu

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Pole představuje nejbližší obraz operační paměti
- Koncepce pole v jazyce Pascal – vždy statické, jednorozměrné
- Složkou pole může být libovolný typ bez omezení
- Přístup ke složce pomocí indexu
- Indexový výraz:

$$A = A_0 + (I - I_0) \cdot V,$$

kde  $A$  je výsledná adresa,  $A_0$  je počáteční adresa pole,  $I$  je aktuální index,  $I_0$  je index prvního prvku pole,  $V$  je velikost složky pole v bytech.

- Pole představuje nejbližší obraz operační paměti
- Koncepce pole v jazyce Pascal – vždy statické, jednorozměrné
- Složkou pole může být libovolný typ bez omezení
- Přístup ke složce pomocí indexu
- Indexový výraz:

$$A = A_0 + (I - I_0) \cdot V,$$

kde  $A$  je výsledná adresa,  $A_0$  je počáteční adresa pole,  $I$  je aktuální index,  $I_0$  je index prvního prvku pole,  $V$  je velikost složky pole v bytech.

- Optimalizace indexového výrazu:
  - a)  $I_0 = 0$ ,
  - b)  $V = 2^n$ , příp.  $V = 1$

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Implementován jako pole `array [0..L] of char`

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Implementován jako pole `array [0..L] of char`
- První složka představuje okamžitou délku

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Implementován jako pole `array [0..L] of char`
- První složka představuje okamžitou délku
- Indexy od 1 do L reprezentují jednotlivé znaky

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Implementován jako pole **array** [0..L] **of** char
- První složka představuje okamžitou délku
- Indexy od 1 do L reprezentují jednotlivé znaky
- Deklarovaná délka L je max. 255



Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Implementován jako pole `array [0..L] of char`
- První složka představuje okamžitou délku
- Indexy od 1 do L reprezentují jednotlivé znaky
- Deklarovaná délka L je max. 255
- Odvozen od typu `char` – jednobytové kódování

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Implementován jako pole `array` `[0..L]` `of` `char`
- První složka představuje okamžitou délku
- Indexy od 1 do L reprezentují jednotlivé znaky
- Deklarovaná délka L je max. 255
- Odvozen od typu `char` – jednobytové kódování
- Řada operací: okamžitá délka, porovnání, hledání, vkládání a mazání podřetězce, zřetězení

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný heterogenní typ

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný heterogenní typ
- Široké použití pro reprezentaci dat (databáze, dynamické struktury)

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný heterogenní typ
- Široké použití pro reprezentaci dat (databáze, dynamické struktury)
- Přístup ke složce prostřednictvím tečkové notace

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný heterogenní typ
- Široké použití pro reprezentaci dat (databáze, dynamické struktury)
- Přístup ke složce prostřednictvím tečkové notace
- Usnadnění přístupu – příkaz `with`

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný heterogenní typ
- Široké použití pro reprezentaci dat (databáze, dynamické struktury)
- Přístup ke složce prostřednictvím tečkové notace
- Usnadnění přístupu – příkaz `with`
- Variantní záznam – některé složky mohou sdílet stejné místo v paměti, používají se výlučně



Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný heterogenní typ
- Široké použití pro reprezentaci dat (databáze, dynamické struktury)
- Přístup ke složce prostřednictvím tečkové notace
- Usnadnění přístupu – příkaz `with`
- Variantní záznam – některé složky mohou sdílet stejné místo v paměti, používají se výlučně
- Operace se záznamem jako celkem nejsou (jen přiřazení identických záznamů)

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole
- Prvek je indexem

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole
- Prvek je indexem
- Hodnota je bit v poloze 0 – nepřítomný prvek, 1 – přítomný prvek

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole
- Prvek je indexem
- Hodnota je bit v poloze 0 – nepřítomný prvek, 1 – přítomný prvek
- Bázovým typem je ordinální typ s hodnotami v intervalu ordinálních čísel 0 až 255

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole
- Prvek je indexem
- Hodnota je bit v poloze 0 – nepřítomný prvek, 1 – přítomný prvek
- Bázovým typem je ordinální typ s hodnotami v intervalu ordinálních čísel 0 až 255
- Free Pascal: dvě velikosti, 4B a 32B



Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole
- Prvek je indexem
- Hodnota je bit v poloze 0 – nepřítomný prvek, 1 – přítomný prvek
- Bázovým typem je ordinální typ s hodnotami v intervalu ordinálních čísel 0 až 255
- Free Pascal: dvě velikosti, 4B a 32B
- Operace – sjednocení, průnik, rozdíl, zjištění přítomnosti prvku

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Specifický typ jazyka Pascal
- Implementována jako bitové pole
- Prvek je indexem
- Hodnota je bit v poloze 0 – nepřítomný prvek, 1 – přítomný prvek
- Bázovým typem je ordinální typ s hodnotami v intervalu ordinálních čísel 0 až 255
- Free Pascal: dvě velikosti, 4B a 32B
- Operace – sjednocení, průnik, rozdíl, zjištění přítomnosti prvku
- Užitečná pomůcka při přetypování

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám
- Princip přístupu k souboru – buffer v paměti

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám
- Princip přístupu k souboru – buffer v paměti
- Rozdělení souborů – textové, netextové; operace jsou rozdílné

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám
- Princip přístupu k souboru – buffer v paměti
- Rozdělení souborů – textové, netextové; operace jsou rozdílné
- Netextové soubory mohou nebo nemusí mít udán typ datové složky



Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám
- Princip přístupu k souboru – buffer v paměti
- Rozdělení souborů – textové, netextové; operace jsou rozdílné
- Netextové soubory mohou nebo nemusí mít udán typ datové složky
- Textové soubory mají čistě sekvenční charakter

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám
- Princip přístupu k souboru – buffer v paměti
- Rozdělení souborů – textové, netextové; operace jsou rozdílné
- Netextové soubory mohou nebo nemusí mít udán typ datové složky
- Textové soubory mají čistě sekvenční charakter
- Netextové soubory umožňují přímý přístup ke složkám, jde o obdobu pole

Datový typ

Jednoduché typy

Strukturované typy

Pole

Znakový řetězec

Záznam

Množina

Soubor

- Jediný datový typ, jehož proměnné nejsou uloženy v operační paměti
- Veškeré manipulace jsou řešeny podprogramy – jazykové rozhraní k systémovým službám
- Princip přístupu k souboru – buffer v paměti
- Rozdělení souborů – textové, netextové; operace jsou rozdílné
- Netextové soubory mohou nebo nemusí mít udán typ datové složky
- Textové soubory mají čistě sekvenční charakter
- Netextové soubory umožňují přímý přístup ke složkám, jde o obdobu pole
- Operace: Assign, reset, rewrite, read, readln, write, writeln, eof, eoln, Close, Seek, FilePos, FileSize