УуБ  Универзитет у Београду

ФОН  Факултет организационих наука

Ξ Катедра за софтверско инжењерство

лабСИ  Лабораторија за софтверско инжењерство

πα  ПРОГРАМИРАЊЕ I

ПРОГРАМИРАЊЕ I

Предавање #2

Проф. др **Саша Д. Лазаревић**, дипл. инж. инф.
sasa.lazarevic@fon.bg.ac.rs

Садржај предавања

1) Основе програмског језика Ц
2) Типови података у ПЈ Ц
3) Литерали
4) Конверзија вредности различитих типова података
5) Изрази и оператори

Садржај предавања

1) **Основе програмског језика Ц**
2) Типови података у ПЈ Ц
3) Литерали
4) Конверзија вредности различитих типова података
5) Изрази и оператори

Основе програмског језика Ц

- Карактеристике
- Верзије и стандарди
- Елементи
- Структура програма
- Изворне датотеке
- Област важења идентификатора

Карактеристике програмског језика Ц

- Виши програмски језик (по структурираним типовима података и управљачким структурама), али са особинама машински зависних програмских језика (по манипулацији битовима, коришћењу процесорских регистара, приступу подацима помоћу адресе и хардверски оријентисаним операторима)
- Језик опште намене, са посебним погодностима за системско програмирање
- Императивни језик
- Модуларни језик
- Блок-структурни језик
- Необјектни језик
- Компактан језик (мали скуп резервисаних речи)
- Ефикасан језик (мали хардверски захтеви, велика брзина)
- Стандардизован и преносив језик
- *Case-sensitive* и *free-form* језик

Верзије и стандарди програмског језика Ц

- ***C*** – Пробитни Ц, Др Денис Ричи, Белове лабораторије, 1972.
- ***K&R C*** – Керниганов и Ричијев Ц, заснован на Референтном приручнику из књиге: *The C Programming Language*; прво издање 1978, друго издање 1988.
- ***C89*** – по националном стандарду *ANSI X3.159:1989* (назван: *ANSI C*), усвојен је међународни стандард *ISO/IEC 9899:1990* + додатак (*Normative Addendum I*): *ISI/IEC 9899/AMDI:1995*.
- ***C99*** – међународни стандард *ISO/IEC 9899:1999* + исправке и допуне (*Technical Corrigenda*): *TC1:2001* и *TC2:2004*.
- ***C11*** – међународни стандард *ISO/IEC 9899:2011*.

Елементи програмског језика Ц 1/5

Скуп знакова (*character set*) – мала и велика слова енглеске абецеде (a – z, A – Z), десет декадних цифара (0 - 9), двадесет девет знакова интерпункције:

```
! " # % & ' * ( ) + - / : ; ^ < > ~ , ? = [ ] _ { . \
| }
```

- Дирафи:
- Триграфи:

Дираф	Еквивалент
<:	[
:>	]
<%	{
%>	}
%:	#
%:%:	##

Триграф	Еквивалент
??(	[
??)	]
??<	{
??>	}
??=	#
??/	\
??!	|
??'	^
??-	~

Елементи програмског језика Ц 2/5

- Лексички симбол = лексема = токен (*token*) – недељиви низ знакова, најмања програмска семантичка јединица. Лексеме се деле на: идентификаторе, константе, резервисане речи, операторе и сепараторе
- Бели знакови (*white spaces*) – размак, хоризонтална табулација, вертикална табулација, нови ред, нова страна
- Коментари (*comments*) –
  - Вишередни или блок коментар: /* */
  - Једноредни коментар: //

Елементи програмског језика Ц 3/5

- Искази (*statements*) – низови лексичких симбола. Деле се на:
  - Декларативне исказе (*declarativ st.*) којима се декларишу делови програма (подаци, потпрограми итд)
  - Извршне (егзекутивне) исказе (*executive st.*) = наредбе којима се одређују акције програма
- Програм (*program*) – низови декларативних и извршних исказа којима се специфицира начин обраде података
- Директиве препроцесора (*preprocessor directives*) – упутства помоћу којих може да се утиче на ток превођења програма. Оне нису део програма, па се зато не убрајају у исказе. Њих узима у обзир претпроцесор, који врши припрему изворног кôда пре самог превођења. Једна директива – један ред: свака директива пише се у засебном реду и мора да почиње знаком #. Не смеју да се мешају са исказима програма

Елементи програмског језика Ц 4/5

- Идентификатори (*identifiers*) – служе за означавање (именовање) свих врста елемената програма: података , симболичких константи, типова података које дефинише програмер, потпрограма и ознака које служе као одредиште за наредбе безусловних скокова.
- Идентификатори могу да се састоје од слова, цифара и доње црте (_). Први знак не сме да буде цифра
- Прави се разлика између великих и малих слова:

  procenat ≠ Procenat ≠ PROCENAT
- Идентификатор може бити произвољне дужине, али се најмање првих 31 знакова користи за идентификацију

Елементи програмског језика Ц 5/5

Резервисане речи (*keywords*) – има их 37 и не смеју се користити као идентификатори:

auto	double	inline	sizeof	volatile
break	else	int	static	while
case	enum	long	struct	_Bool
char	extern	register	switch	_Complex
const	float	restrict	typedef	_Imaginary
continue	for	return	union	
default	goto	short	unsigned	
do	if	signed	void	

Структура програма 1/2

Функција (*function*) – основни градивни елемент програмског језика Ц; састоји се из секвенце исказа; унутар функције се наредбе (тј. извршни искази) могу груписати у блокове (*blocks*)

- Две врсте функција:
  - Готове функције из стадардне библиотеке - прототипови ф-ја из стандардне библиотеке налазе се у датотекама заглавља (*header files, *.h*)
  - Програмерски креиране функције, налазе се у датотекама изворног кода (*source code files, *.c*); могу да се налазе у програмерски креираним библиотекама *(*.h + *.obj*)

Структура програма 2/2

- Сваки програм има једну обавезну функцију: *main()*
  - Ова ф-ја се прва позива; представља највиши ниво управљања током извршења програма
- Дефиниције функција не могу се угнездити једна у другу, тј. не постоје подфункције
- Функције се могу дефинисати произвољним редоследом
- Функције могу међусобно да се позивају; функција може и саму себе да позива (рекурзија)
- Декларација функције ≠ дефиниција функције
- Пре позива функције мора да претходи њена или декларација или дефиниција

За структуру програма видети примере:
од P02_01_krug.c до P02_06_krug.c

Изворне датотеке 1/2

У ПЈ Ц постоје две врсте изворних датотека:

1) *.c – изворне датотеке, садрже програмски кôд
2) *.h – датотеке заглавља, садрже прототипове функција, дефиниције константи и дефиниције макроа

Уобичајена структура *.c датотеке:

1) Претпроцесорске директиве
2) Глобалне декларације
3) Дефиниције функција

Изворне датотеке 2/2

Начин укључивања *.*h* датотеке у изворни код остварује се употребом директиве *#include*

Укључивање *.*h* датотеке из системског *include*-директоријума:

```
#include <stdio.h>
```

Укључивање *.*h* датотеке из текућег директоријума:

```
#include "P02_krug.h"
```

Изворна датотека на језику Ц, заједно са свим датотекама заглавља које садржи, сачињава **јединицу за превођење** *(translation unit)*. Преводилац (*compiler*) преводи њен садржај секвенцијално, разлажући изворни код на лексеме (токене)

Област важења идентификатора 1/2

Област важења (*scope*) идентификатора – део јединице за превођење у којем идентификатор има значење. Другим речима, то је онај део програма у којем је идентификатор доступан („видљив“)

Врсту области важења увек одређује место на којем се идентификатор декларише.

Напомена: Изузетак су лабеле, јер је њихова област важења увек функција

Област важења идентификатора 2/2

Постоје четири врсте области важења:

1. Област датотеке (*file scope*) – ако се идентификатор декларише изван свих блокова и листа параметара, његова област важења је датотека. Тада се може користити свуда унутар датотеке, почев од места декларације па до њеног краја
2. Област блока (*block scope*) – сви идентификатори декларисани у блоку, изузев лабела, важећи су у том блоку. Декларације се не морају налазити пре наредби у блоку. Нпр. тело дефиниције функције чини један блок
3. Област прототипа функције (*function prototype scope*) – имена параметара у прототипу ф-је припадају области важења тог прототипа. Пошто имена параметара немају значење ван прототипа, корисна су само за подсећање, па се зато могу изоставити
4. Област функције (*function scope*) – област важења лабеле увек је блок ф-је у којој се појављује, чак и када је блок угнежђен

За област важења идентификатора видети пример:
P02_07_oblast.c

Садржај предавања

1) Основе програмског језика Ц
2) **Типови података у ПЈ Ц**
3) Литерали
4) Конверзија вредности различитих типова података
5) Изрази и оператори

Типови података у ПЈ Ц

Логички аспект: Формално, **тип података** је алгебарска структура и одређује се као:

- (непразан) скуп вредности /домен/,
- (непразан) скуп операција над датим скупом вредности /интерфејс/ и
- (могуће празан) скуп константи.

ТИП ПОДАТАКА = СВ + СО + СК

Физички аспект: Материјално, **тип података** је одређен меморијском репрезентацијом:

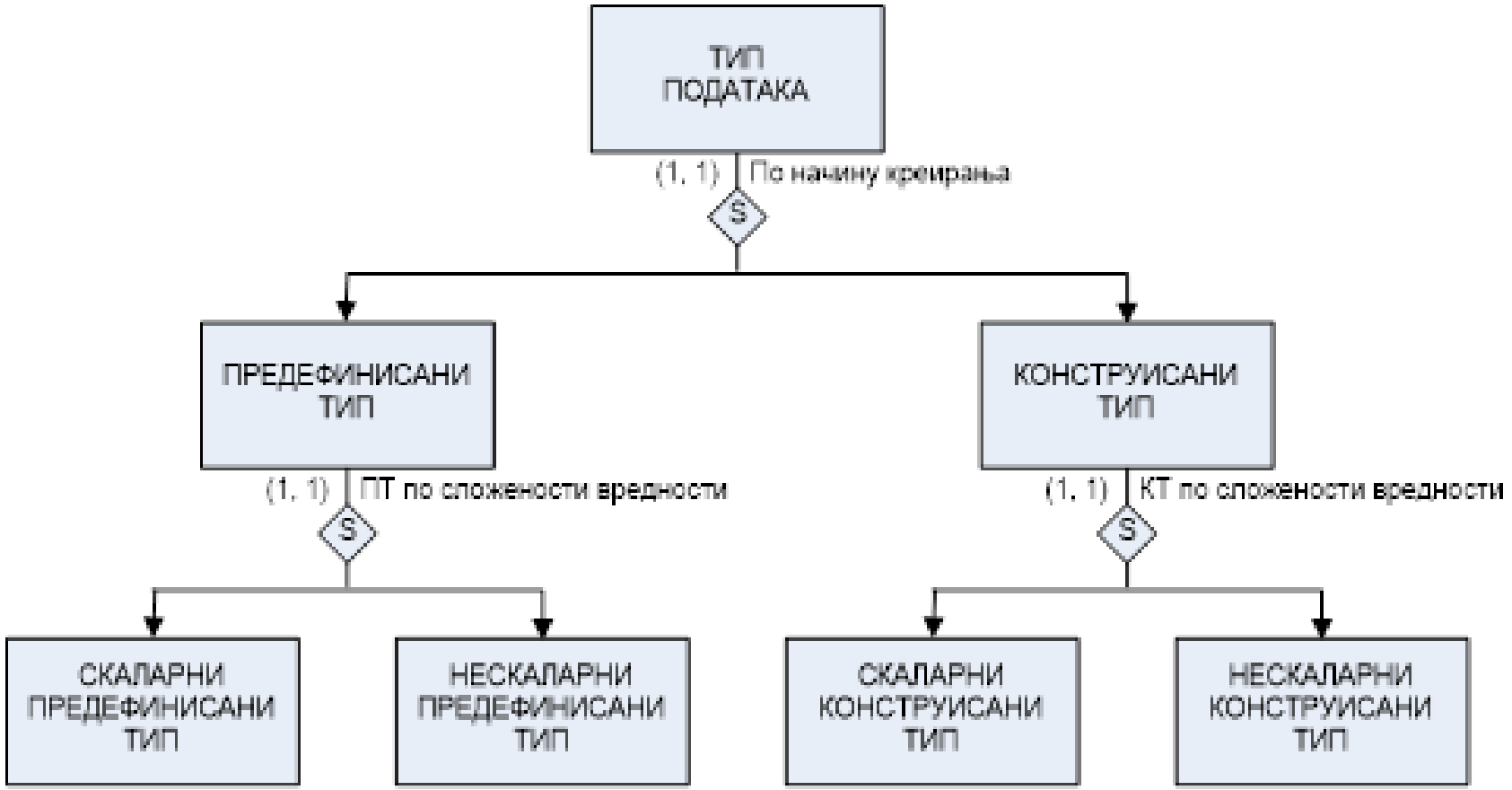
- величином меморије неопходном за меморисање вредности типа, исказана у *By*
- начин кодирања вредности типа

Типови података у ПЈ Ц

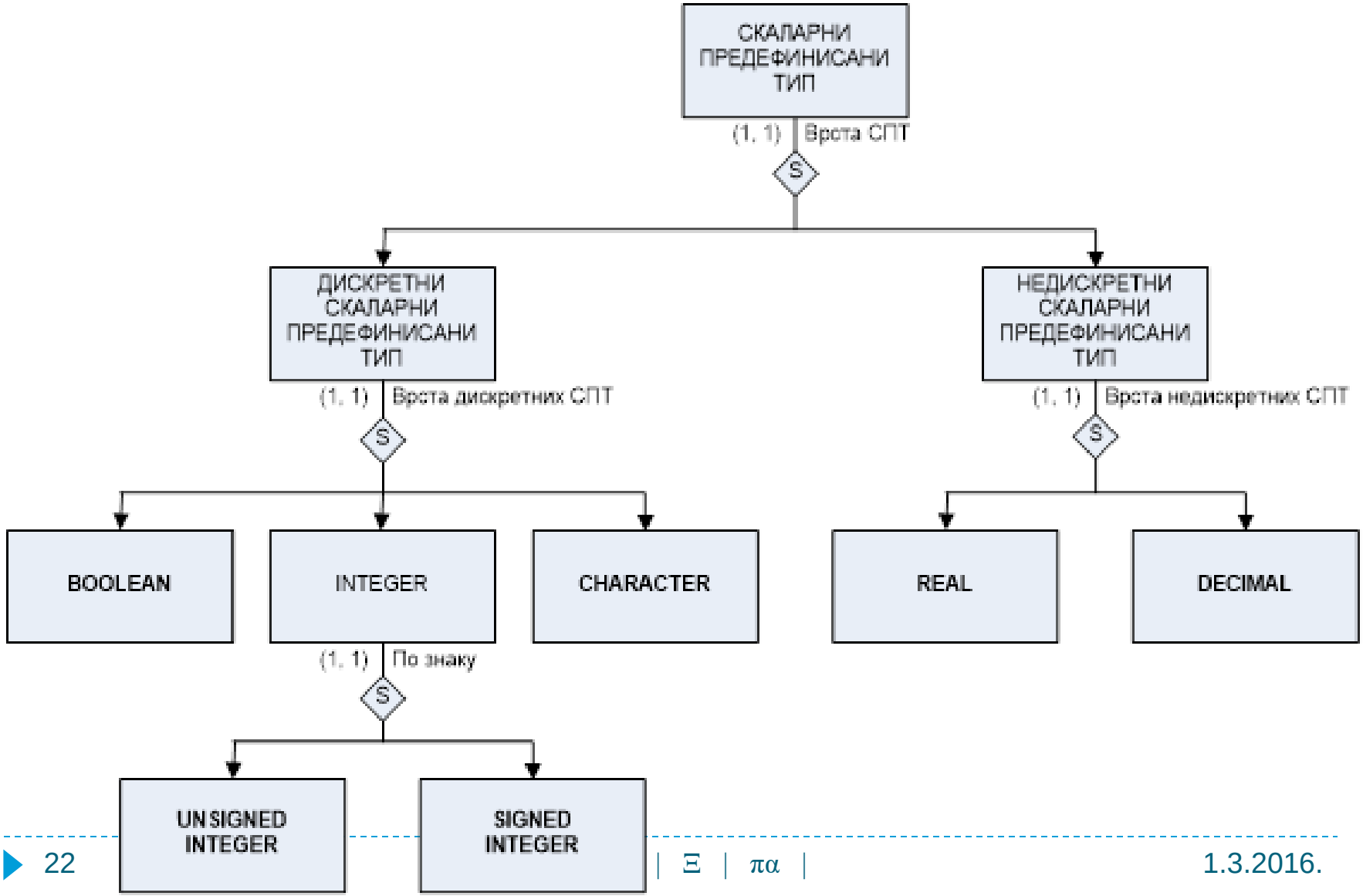
На пример, скуп целих означених бројева int (синоними: signed, signed int) има:

- Домен: [-2.147.483.648, 2.147.483.647]
- Операторе: +, -, /, %, * итд.
- Константе: INT_MIN и INT_MAX
- Величину: 4 By

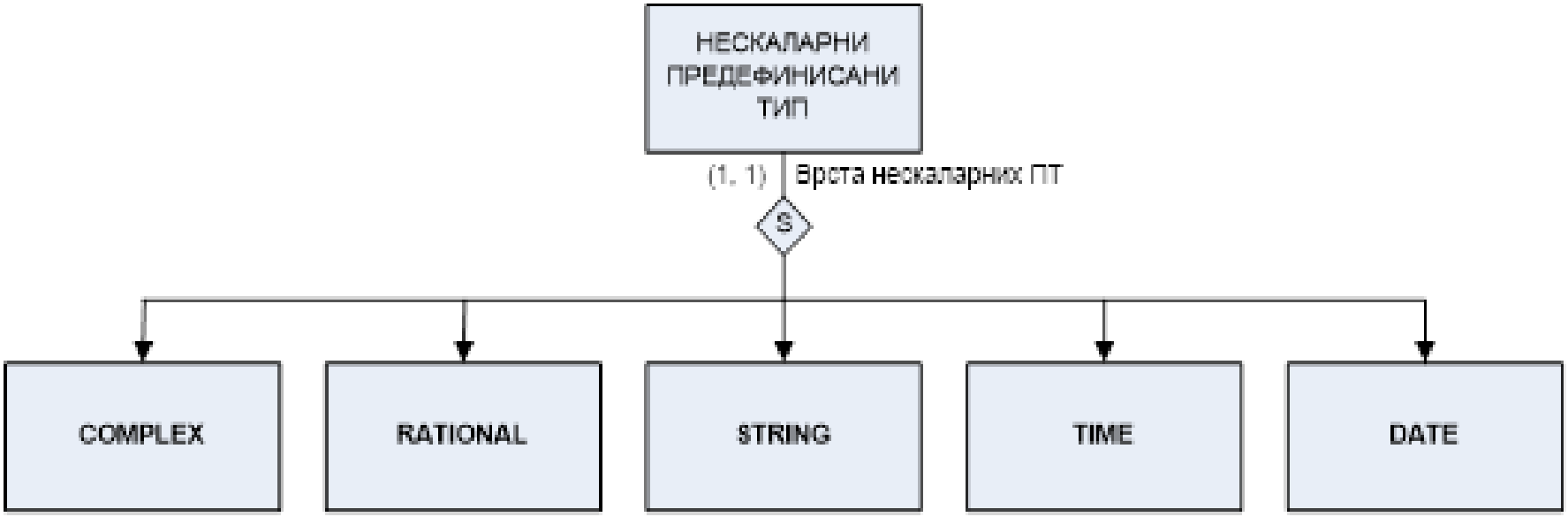
Класификација типови података у ПЈ



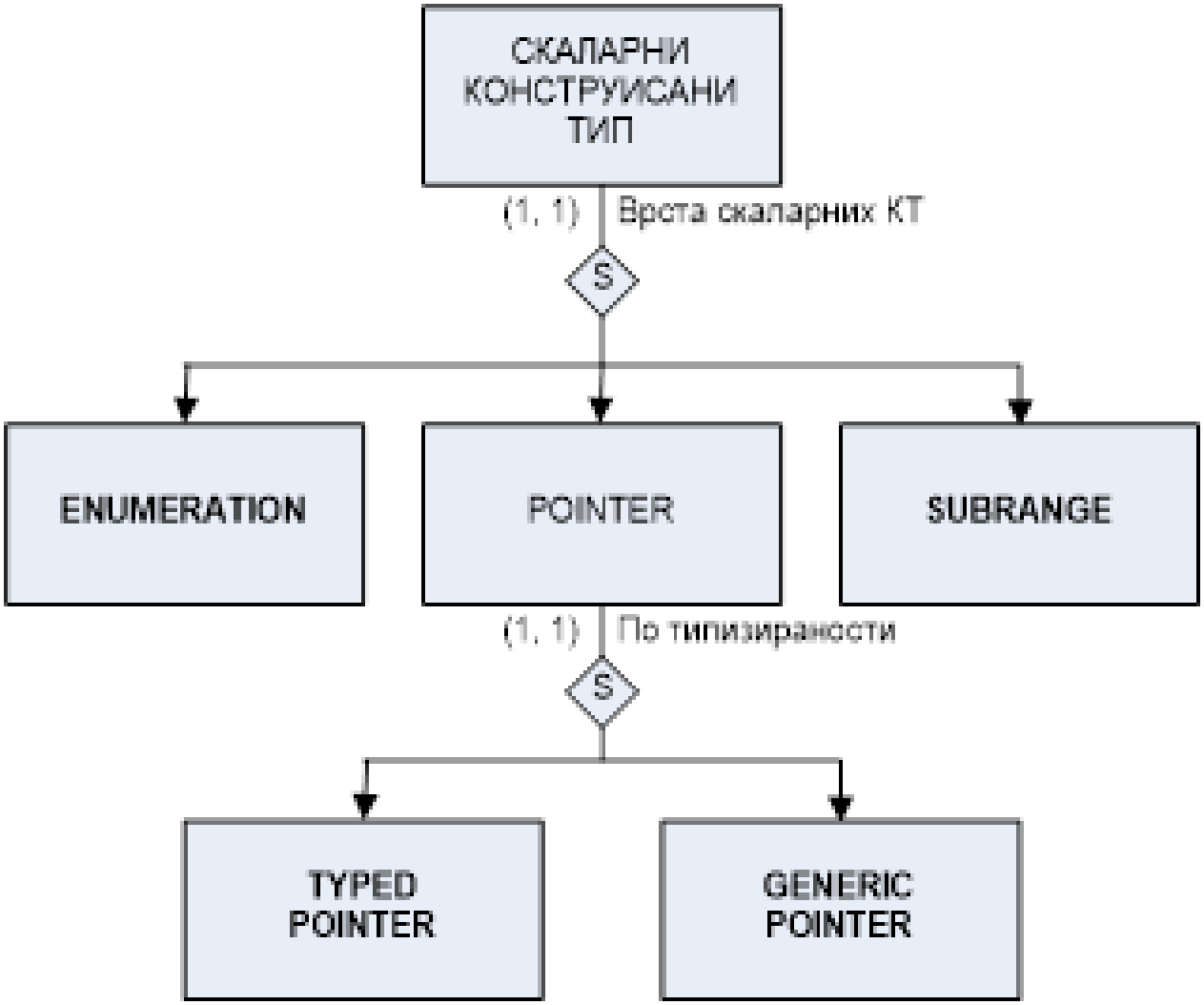
Класификација типови података у ПЈ



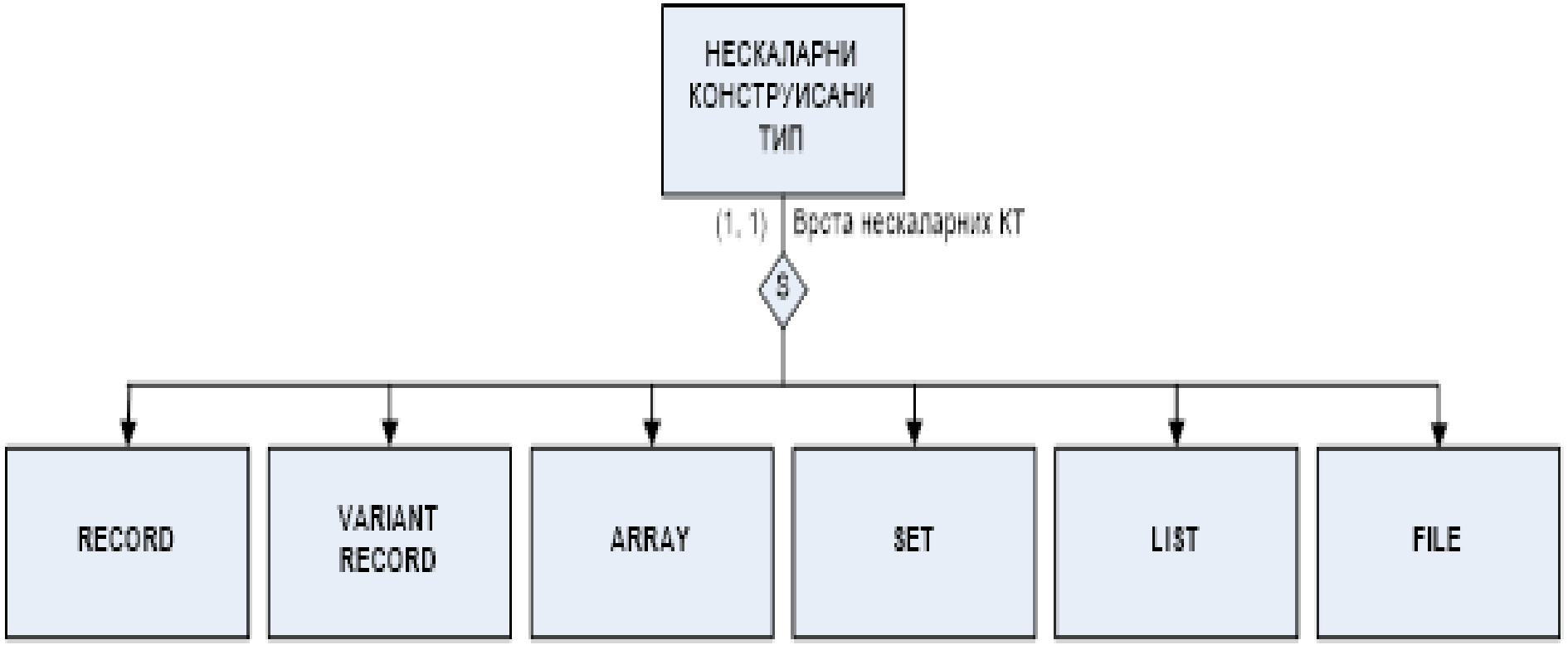
Класификација типови података у ПЈ



Класификација типови података у ПЈ



Класификација типови података у ПЈ



Типови података у ПЈ Ц

У ПЈ Ц, појам **објекат** означава место у меморији чији садржај може да представља вредност.

- Објекти који имају имена називају се **променљиве** (*variables*)
- Објекти који имају имена и додељену вредност која се не може мењати називају се **константе** (*constants*)
- Тип објекта одређује колико простора објекат заузима у меморији и како се кодирају његове могуће вредности.

  На пример, исти низ битова може представљати потпуно различите целе бројеве, зависно од тога да ли се тумачи као означена (signed) или неозначена (unsigned) вредност
- ПЈ Ц познаје само нумеричке типове података

Целобројни типови података у ПЈ Ц

- Основни целобројни тип: ***int***
- Спецификатори дужине: *short* и *long*
- Спецификатори знака: *unsigned* и *signed*
- Уобичајена величина: *2By* или *4By*
- Константе (у *limits.h*):
  - SHRT_MAX, INT_MAX, LONG_MAX, LLONG_MAX
  - SHR_MIN, INT_MIN, LONG_MIN, LLONG_MIN

sizeof(short) ≤ sizeof(int) ≤ sizeof(long) ≤ sizeof(long long)

Целобројни типови података у ПЈ Ц

- Најмањи целобројни тип: ***char***
- Спецификатори: *unsigned* и *signed*
- Уобичајена величина: *1By*
- Константе:
  - CHAR_MAX, SCHAR_MAX, UCHAR_MAX
  - CHAR_MIN, SCHAR_MIN

```
sizeof(unsigned char) = sizeof(char)
                      = sizeof(signed char)
```

Целобројни типови података у ПЈ Ц

- Највећи целобројни тип: ***long long int***
- Спецификатори: *unsigned* и *signed*
- Уобичајена величина: *8By*
- Константе:
  - LLONG_MAX, ULLONG_MAX
  - LLONG_MIN

```
sizeof(unsigned long long) = sizeof(signed long long)
```

Реални типови података у ПЈ Ц

- Реални ТП = ТП у формату са покретним зарезом (*floating-point data type*)
- Три врсте:
  - *float* (*4 By*, једнострука прецизност, 6 значајних цифара, опсег вредности: 1.17×10^{-38} до $3.40 \times 10^{+38}$),
  - *double* (*8 By*, двострука прецизност, 15 значајних цифара, опсег вредности: 2.22×10^{-308} до $1.79 \times 10^{+308}$) и
  - *long double* (*10 By*, проширена прецизност, 19 значајних цифара, опсег вредности: 3.4×10^{-4932} до $1.1 \times 10^{+4932}$)

```
sizeof(float) < sizeof(double) < sizeof(long double)
```

Логички тип података у ПЈ Ц

- У стандарду *C99* уведен је неозначени целобројни тип *_Bool* за представљање логичких вредности
- Нула је нетачно = *false*, а свака ненулта вредност је тачно = *true*
- Уколико се у програм укључи заглавље *stdbool.h* могу се користити идентификатори:
  - bool (макро, синоним за *_Bool*),
  - true (симболичка константа, вредност 1) и
  - false (симболичка константа, вредност 0)

Комплексни тип података у ПЈ Ц

- Стандард *C99* подржава математичке операције са комплексним бројевима
- Комплекасан број је...
- Три врсте:
  - `float _Complex`
  - `double _Complex`
  - `long double _Complex`
- Пример:

```
#include <complex.h>
double complex  cmpx = 2.1 + 3.2 * I;
```

Набројиви тип података у ПЈ Ц

- Набројиви тип или енумерација (*enumeration, enumerated type*) је целобројни тип који програмер сам дефинише у програму:

```
enum boja { CRVENA, BELA, PLAVA };
```

- Енумерационе константе *CRVENA, BELA, PLAVA* имају вредности 0, 1 и 2, респективно. Могућа је и додела сопствених вредности:

```
enum boja { CRVENA, BELA, PLAVA, SIVA = 8, BRAON};
```

или додељивање истих вредности:

```
enum boja {
    OFF, ON,
    STOP = 0, GO = 1,
    CLOSED = 0, OPEN = 1 };
```

Тип података ***void*** у ПЈ Ц

- Врло посебан тип. Зашто? Зато што ***нема познатих вредности тог типа***. Зато се не може декларисати променљива или константна овога типа
- Где се може користити тип *void*:
  1) У декларацији функција:
     ```
     int main (void){...}
     int TekucaGodina(void);
     ```
  2) За изразе типа *void*:
     ```
     void PrikaziPozdrav(char * txt);  // procedura
     (void)printf("Ne treba mi povratna vred. ove f-je\n");
     ```
  3) За генеричке показиваче (показивачи на тип *void*):
     ```
     void * podatak;
     void *malloc(size_t size);
     void free(void *ptr);
     ```

Садржај предавања

1) Основе програмског језика Ц
2) Типови података у ПЈ Ц
3) **Литерали**
4) Конверзија вредности различитих типова података
5) Изрази и оператори

Литерали

- Литерал је лексема која означава непроменљиву вредност. Та вредност може бити број, знак или ниска (низ знакова, знаковни низ). Тип литерала одређен је његовом вредношћу и начином записа
- Разликују се:
  - Целобројни литерали:
    - Декадни,
    - Октални и
    - Хексадекадни.
  - Реални литерали:
    - Декадни и
    - Хексадекадни
  - Знаковни литерали:
    - ANSI (једнобајтни),
    - Вишебајтни и
    - Управљачки знаци
  - Ниске (стринг) литерали (литерали типа знаковних низова)

Целобројни литерали

- Целобројни литерали - цели бројеви, чија се бројевна основа задаје префиксом
- Основе:
  - Декадна - цели бројеви који почињу цифром другачијом од 0: 39, 255, -673...
  - Октална - цели бројеви који почињу цифром 0: 047 = $(39)_{10}$, 0377 = $(255)_{10}$, 0176537 = $(-673)_{10}$ ...
  - Хексадекадна – цели бројеви који почињу префиксом *0x* или *0X*: 0x27 = $(39)_{10}$, 0xFF = $(255)_{10}$, 0xFD5F = $(-673)_{10}$ ...
- Целобројни литерали са суфиксом:
  - 515U unsigned int
  - 0L long
  - 0xF0BUL unsigned long (основа хексадекадна)
  - 0777LL long long (основа октална)
  - 0xAAALLU unsigned long long (основа хексадекадна)

Реални литерали

- Реални литерали: реални бројеви, чија се бројевна основа задаје префиксом
- Основе:
  - Декадна - релни бројеви који почињу цифром другачијом од 0: 22.0, 3.45E6, 67e-8, ...
  - Хексадекадна – реални бројеви који почињу префиксом *0x* или *0X*
- Може: .123 и 12. и .234E5
- Реални литерали са суфиксом – подразумевани тип реалних литерала (литерала у формату са покретним зарезом) је *double*. Додавањем суфикса *F* или *f*, литералу се додељује тип *float*; а суфиксом L или l, литералу се додељује тип *long double*. На пример:

```
float fltProm = 456.789F;
long double ldbProm = 7890.1L;
```

Знаковни литерали

- Знаковни литерали се састоје од једног или више знакова под једноструким наводницима. На пример: 's', 'ASA', '9', '*', '=====', …
- Изузетак у начину приказивању су знакови:
  - Апостроф '\'' је '
  - Обрнута коса црта '\\' је \
  - Управљачки знак за нови ред '\n' је нови ред
- По величини, тј. по броју бајтова, постоје:
  - ANSI (једнобајтни) знакови и
  - Вишебајтни знакови

Знаковни литерали

- Управљачки знаци почињу обрнутом косом цртом и представљају један знак
- Важнији упр.знаци:
  - `\'` полунаводник
  - `\''` наводник
  - `\?` знак питања
  - `\\` обрнута коса црта
  - `\a` упозорење (*alert*)
  - `\b` брисање уназад (*backspace*)
  - `\f` нова страна (*form feed*)
  - `\n` нови ред (*new line*)
  - `\r` почетак реда (*carriage return*)
  - `\t` хоризонтални табулатор
  - `\v` вертикални табулатор

Литерали типа знаковних низова

- Ниске (стрингови) литерали или литерали типа знаковних низова су низови знакова под наводницима. На пример:
  "Volim programiranje!\n"
- Исписати на стд.излаз.уређају текст:

  Preuzmite izvorni kod iz foldera ".\FON\P1\kod"

  Имплементација у ПЈ Ц:

```
...
char code_path[128] = ".\\FON\\P1\\kod";
printf("Preuzmite izvorni kod iz foldera \"%s\" \n“, code_path);
...
```

- Вишередни текст:

```
char *info = "Ovo je tekst\
 napisan \n u tri\
 reda, a bice prikazan u dva. Zasto\?\n";
```

Садржај предавања

1) Основе програмског језика Ц
2) Типови података у ПЈ Ц
3) Литерали
4) **Конверзија вредности различитих типова података**
5) Изрази и оператори

Конверзија вредности различитих типова података

- За домаћи

Садржај предавања

1) Основе програмског језика Ц
2) Типови података у ПЈ Ц
3) Литерали
4) Конверзија вредности различитих типова података
5) **Изрази и оператори**

Изрази и оператори

- За домаћи

Шта ћете радити на вежбама

1) Пример: оператори и изрази
2) Пример: селекција
   - if, if-else, if-else-if…
   - switch
3) Пример: итерација
   - for
   - while
   - do-while
4) Задаци:
   - Највећи број
   - Провера парности броја
   - Факторијел

Следеће предавање

Наредбе:

1) Израз
2) Блок
3) Избор: селекција или разграната управљачка структура
4) Понављање: итерација или циклична управљачка структура
5) Скок: безусловни пренос тока извршења наредби

УуБ  Универзитет у Београду

ФОН  Факултет организационих наука

Ξ Катедра за софтверско инжењерство

лабСИ  Лабораторија за софтверско инжењерство

πα  ПРОГРАМИРАЊЕ I

ПРОГРАМИРАЊЕ I

Предавање #2

Проф. др **Саша Д. Лазаревић**, дипл. инж. инф.
sasa.lazarevic@fon.bg.ac.rs

Универзитет,
факултет,
катедра,
лабораторија,
предмет