

Дәріс-8. Java ортасында тіркестерді өңдеу жолдары

Жоспар:

- 1 Конструкторлар және тіркестерді меншіктеу
- 2 Тіркестердің бөлігін қосу және меншіктеу
- 3 Тіркестерді түрлендіру
- 4 Тіркестердің құрамын іздеу
- 5 Тіркестердің бөліктерін салыстыру
- 6 Тіркестердің сипаттамаларын алу

1 Java ортасындағы сөз тіркестері

Java тілінде арнайы сөз тіркесіне арналған тип қарастырылмаған. Бірақ Java ортасында тіркестермен жұмыс істеудің екі мүмкіндігі бар: C кітапханасынан қалған (`<string.h>` не `<cstring>`) тақырып файлында сипатталған функциялар және Java-тің `string` кітапханалық класы. C-дан мұрагерлікке қалған функцияларды қолдану ыңғайсыз және қауіпті, себебі олар сөз тіркестерінің шекарадан шығып кетуін қадағалай алмайды.

Стандартты кітапхананың `string` типі тіркестердің ақпараттың шекарасынан шығып кетуінен қорғалған және олар операциялар мен тәсілдер көмегімен жұмыс істей алады.

Енді тіркестермен жұмыс істеудің негізгі ерекшеліктері мен тәсілдерін қарастырайық. Сөз тіркесі немесе тіркестер – нөл символмен аяқталатын символдар жиымы (массиві).

Олардың ұзындығы қажеттілікке қарай динамикалық түрде өзгертіліп отырады. Тіркестермен жұмыс істеу класын пайдалану үшін программаға `<string>` тақырыптық файлын енгізу керек. Бұл тақырыптық файл бұрынғы шыққан компиляторларда `<cstring.h>` (`<bstring.h>`, `<string.h>`) түрлерінде де кездесуі мүмкін.

2 Конструкторлар және тіркестерді меншіктеу

`String` класында бірнеше конструкторлар анықталған, олардың жиі қолданылатындары: `string ()`;

`string (const char *)`; `string (const char *, int n)`; `string (string &)`;

Бірінші конструктор `string` типті бос объект құрады. Екінші конструктор ескі стильдегі тіркестер негізіндегі `string` типті объект құрады, ал үшіншісі `string` типті объект құрады да, бірінші параметрде көрсетілген тіркестен `n` символды алып осыған жазады. Төртінші конструктор көшіру конструкторы болып табылады, ол параметр ретінде берілген объектінің көшірмесі арқылы жаңа объектіні құрады.

`string` класында үш меншіктеу операциясы анықталған: `string&operator= (const string&str)`;

`string&operator= (const char*s)`; `string&operator= (char c)`;

Мұнан бір тіркеске `string` типтегі басқа тіркесті немесе жеке символды да меншіктеуге болатыны көрініп тұр. Мысалы:

```
string s1;  
string s2("Azat"); string s3(s2); s1='X'; s1="Azat"; s2=s3;
```

string класының функциялары: тіркестің бөлігін қосу және меншіктеу, тіркестерді түрлендіру, ішкі тіркесті іздеу, тіркес бөліктерін салыстыру.

Тіркес элементін анықтау үшін бұрынғы индекстеу операциясы қолданылады және at функциясы анықталған. Мысалы:

```
string s("Azat"); cout << s.at(1);
```

Егер индекс тіркес ұзындығынан артып кетсе, онда out_of_range аластау (исключение) оқиғасы туындайды.

3 Тіркес бөлігін қосу және меншіктеу

Бір тіркестің ішкі бөлігін екінші бір тіркеске меншіктеу үшін assign функциясы қолданылады:

```
assign (const string &str);
```

```
assign (const string &str, size_type pos size_type n); assign (const char *s,  
size_type n);
```

Алғашқы форма str тіркесін оны шақырған тіркеске меншіктейді, бұл функция әрекеті меншіктеу операциясына эквивалентті болып саналады. Мысалы:

```
string s1("Azat"), s2  
s2.assign (s1); // бұл s2=s1 дегенмен бірдей
```

Жоғарыдағы функциялардың екінші формасы шақырушы тіркеске str тіркесінің pos позициясынан басталған n символдардан тұратын бөлігін меншіктейді. Егер pos+n мәні тіркес ұзындығынан артық болса, онда str тіркесі толық меншіктеледі. Функциялардың үшінші формасы шақырушы тіркеске s тіркесінің n символын меншіктейді.

Мысалы:

```
string s1("telefon"), s2("olar"); s1.assign (s2); // нәтижесі: s1=s2=olar  
s1.assign (s2, 1, 3); // нәтижесі: lar
```

Бір тіркестің бөлігін екінші тіркеске қосып біріктіру үшін append функциясы қолданылады:

```
append (const string&str);
```

```
append(const string&str, size_type pos, size_type n); append (const char*s,  
size_type n);
```

Бірінші форма str тіркесін шақырушы тіркестің соңына қосады, оның әрекеті конкатенация амалына (біріктіруге) ұқсас болып келеді. Екінші форма шақырушы тіркеске pos позициясынан басталған str тіркесі бөлігін меншіктейді. Мысалы: string s1("fon"), s2("dar");

```
s1.append (s2); // нәтижесі: fondar s1.append (s2, 1, 2); // нәтижесі: fonar
```

4 Тіркестерді түрлендіру

Бір тіркес бөлігін екінші тіркеске кірістіріп қою (вставка) үшін insert функциясы қолданылады:

```
insert (size_type pos1, const string &str);  
insert (size_type pos1, const string &str, size_type pos2, size_type n);  
insert (size_type pos, const char *s, size_type n);
```

Бірінші форма str тіркесін оны шақырған тіркеске pos1 позициясынан бастап кірістіріп орналастырады да, одан кейін шақырған тіркестің қалған символдары жалғасады. Егер pos1 тіркес ұзындығынан артық болса, онда out_of_range оқиғасы орын алады.

Екінші формада оны шақырған тіркестің pos1 позициясы сақталады, сонан соң оның pos2 позициясынан бастап str тіркесінің n символы кірістіріледі де, одан кейін шақырған тіркестің қалған символдары жалғасады.

Функцияның үшінші формасында оны шақырған тіркестің pos1 позициясынан бастап бұрынғы типтегі s тіркесінің n элементін кірістіріп орналастырады.

Тіркестің ішкі бөлігін жою үшін erase функциясы қолданылады: erase (size_type pos=0, size_type n=npos);

Ол шақырылған тіркестің pos позициясынан бастап n элементін жояды. Егер позиция көрсетілмесе, онда ол тіркес басынан жойылады.

Бүкіл тіркесті тазалау қызметін clear функциясы атқарады: void clear();

Бақылау сұрақтар:

1. Java тілінде арнайы сөз тіркесіне арналған тип бар ма?
2. Java ортасында тіркестермен жұмыс істеудің қандай мүмкіндігі бар?
3. String класында бірнеше конструкторлар анықталған, олардың жиі қолданылатындары қандай?
4. Бір тіркестің ішкі бөлігін екінші бір тіркеске меншіктеу үшін қандай функция қолданылады?
5. Бір тіркестің бөлігін екінші тіркеске қосып біріктіру үшін қандай функция қолданылады?
6. Бір тіркес бөлігін екінші тіркеске кірістіріп қою (вставка) үшін қандай функция қолданылады?