

3. Учебен програмен език STUDENT

Цел на упражнението

Упражнението представя програмния език STUDENT, с който работи учебният компилатор. Представени са елементите на езика – ключови думи, типове данни, променливи, константи, оператори, изрази, подпрограми. Демонстрирани са примерни програми на STUDENT за решаване на базови задачи.

3.1. Програмен език STUDENT

За целите на учебния компилатор е разработен програмен език, наречен STUDENT.

С цел опростяване на реализацията на компилатора в езика STUDENT са въведени ограничения, най-важните от които са:

- Езикът не е обектно-ориентиран
- Поддържа три прости типа данни (целочислен, булев, символен) и един структуриран тип данни (само едномерен масив)
- Поддържа само подпрограми функции

3.1.1. Азбука

Азбуката на езика включва следните символи:

- малки и главни букви ('a'..'z', 'A'..'Z')
- цифри ('0'..'9')
- символи ('+', '-', '%', ';', '*', '/', ',', '(', ')', '[', ']', '{', '}', '@', '&', '|', '!', '>', '<', '=', '"', "'")

3.1.2. Ключови думи

В езика STUDENT са реализирани 14 ключови думи. Следващият списък представя всички ключови думи и пояснява предназначението на онези от тях, които не се срещат във всички често използвани програмни езици.

- if
- int
- char
- else
- read – прочитане на данни от стандартния вход
- true
- void
- false
- while
- print – извеждане на данни на стандартния изход
- length – брой символи на масив
- return
- boolean

- program – начало на програма на STUDENT

3.1.3. Типове данни, константи, променливи и изрази

- Прости типове данни
 - целочислен тип
 - булев тип
 - символен тип
- Структурирани (съставни) типове данни
 - едномерен масив

При дефиниране на масив се задават тип на данните и брой на елементите, напр. дефиниране на масив от 10 цели числа:

```
int[] arr = int[10];
```

- Константи
 - Целочислени
 - Символни (записват се в апострофи, напр. 'H', 'e', 'l', 'l', 'o')
 - Низови (записват се в кавички, напр. "Hello")
 - Булеви (true и false)
- Променливи
 - глобални
 - локални

Имената на променливите трябва да започват с буква, след която може да следва последователност от букви и/или цифри.

- Изрази
 - аритметични
 - логически
- Оператори
 - аритметични оператори (+, -, *, /, %)
 - логически оператори (&&, ||, !)
 - оператори за сравнение – релационни оператори (<, >, <=, >=, !=, ==)
 - оператор за присвояване (=)
 - унарни оператори (-)
 - условен оператор (if else)
 - оператор за цикъл (while)
 - съставен оператор ({ ... })
 - входно-изходни оператори (read, print)

Съставният оператор в STUDENT обединява два или повече оператори в тялото си. Започва със символ '{' и завършва с '}'.

- Рекурсия
- Подпрограми - функции, които приемат параметри и връщат резултат

При извикването на функция задължително трябва да се използва специалният символ '@' преди името на функцията.

Програма на STUDENT започва с ключовата дума `program`. Тялото на програмата може да включва дефиниции на функции. Функцията се дефинира със списък от формални параметри и се извиква със списък от актуални параметри. В дефиницията на функцията се посочва от какъв тип е резултатът на функцията. Това става с двусимволния оператор “->” след името на функцията и посочване на тип данни (напр. `int`, `char`). Програмата трябва да съдържа поне една функция с име `main` (главна функция). Тя се дефинира задължително без параметри и не връща резултат (тип `void`).

- Управляващи символи (character escaping) - напр. `'\n'`, `'\t'`

3.2. Примерни програми на STUDENT

Пример 1. Програма на STUDENT за извеждане на символен низ “Hello, world!”.

```
program {
    main() -> void {
        print("Hello, world!");
    }
}
```

Пример 2. Програма на STUDENT с функция за намиране на по-голямото от две цели числа, подадени като параметри.

```
program {
    greater(int num1, int num2) -> int {
        if (num1 > num2) {
            return num1;
        }
        else {
            return num2;
        }
    }
    main() -> void {
        print(@greater(10, 5));
    }
}
```

Пример 3. Програма на STUDENT, която извиква функция за събиране на две цели числа, подадени като параметри на функцията.

```
program {
    suma(int a, int b) -> int {
        return a + b;
    }
    main() -> void {
        print("Suma = ", @suma(10, 5));
    }
}
```

Пример 4. Програма на STUDENT за въвеждане на число в диапазона 0-9 от стандартния вход. Програмата демонстрира операторите за въвеждане и извеждане на данни от/на стандартен вход/изход (съответно read и print) и оператора за цикъл while.

```
program {
    main() -> void {
        int a = -1;
        read(a);
        while ( (a < 0) || (a > 9) ) {
            print("Enter a number in the range 0-9: ");
            print("\n");
            read(a);
        }
        print("a = ", a);
        print("\n");
    }
}
```

Пример 5. Програма на STUDENT, която въвежда 10 цели числа в масив и извежда стойностите на елементите на масива на стандартния изход.

```
program {
    main() -> void {
```

```
int[] arr = int[10];
```

```
    arr[0] = 1;
```

```
    arr[1] = 2;
```

```
    arr[2] = 3;
```

```
    arr[3] = 4;
```

```
    arr[4] = 5;
```

```
    arr[5] = 6;
```

```
    arr[6] = 7;
```

```
    arr[7] = 8;
```

```
    arr[8] = 9;
```

```
    arr[9] = 10;
```

```
int i = 0;
```

```
while (i <= 9)
```

```
{
```

```
    print(arr[i]);
```

```
    print(" ");
```

```
    i = i + 1;
```

```
}
```

```
}
```

```
}
```

Контролни въпроси:

1. Кои са ключовите думи в програмния език STUDENT?
2. Какви типове данни поддържа STUDENT?
3. Операторите в езика STUDENT могат да бъдат съставени от един или от два символа. Посочете едносимволните и двусимволните оператори в езика и обяснете тяхното предназначение.
4. Как се извиква функция в езика STUDENT?

Задачи:

1. Да се напише програма на STUDENT, която извежда сумата на целите числа от 1 до 10.
2. Да се напише програма на STUDENT, която извежда стойността на факториел за число, подадено като параметър на функция, като се използва рекурсивен алгоритъм.
3. Да се напише програма на STUDENT, която извежда стойността на факториел за число, подадено като параметър на функция, като се използва итеративен алгоритъм.
4. Да се напише програма на STUDENT, която изчислява и извежда сумата на числата в масив от 10 цели числа.