Функции от по-висок ред

доц. д-р. Нора Ангелова

Функции от по-висок ред

Указател

```
int intValue;   →   int * intPtr;
```

Указател към функция

Тип функция?

Функции от по-висок ред

Указател към функция

<тип_на_функция>(*<указател_към_функция>)(<формални_параметри>)
[=<име_на_функция>]$_{опц}$

- <указател_към_функция> - идентификатор.
- <име_на_функция> - идентификатор, означаващ име на функция от съответния тип.
- <тип_на_функция> и <формални_параметри> - съответстват на тези от дефиницията на функция.

Функции от по-висок ред

Пример:

```
int maxCommonPreffix(const char *str1, const char* str2) {
  int i = 0;
  while(str1[i] && str2[i] && str1[i] == str2[i]) {
    i++;
  }
  return i;
}
...
int (*functionPtr)(const char*, const char*) = maxCommonPreffix;

char str1[10] = "adh123";
char str2[10] = "adhdg";
std::cout << functionPtr(str1, str2); // 3
```

Функции от по-висок ред

Да се напише функция, която намира стойността на

$$\sum_{\substack{i=a,\\ i \to next(i)}}^{b} f(i)$$

Функции от по-висок ред

$$\sum_{\substack{i=a,\\ i \to next(i)}}^{b} f(i)$$

```
double sum(double a, double b, double (*f) (double), double (*next) (double)) {
  double s = 0;
  for(double i = a; i < b + 1e-14; i = next(i)) { // Работим с реални числа
    s = s + f(i);
  }

  return s;
}
```

Функции от по-висок ред

Да се напише функция, която намира стойността на

$$\prod_{\substack{i=a,\\ i \to next(i)}}^{b} f(i)$$

Функции от по-висок ред

$$\prod_{\substack{i=a,\\ i \to next(i)}}^{b} f(i)$$

```
double prod(double a, double b, double (*f)(double), double (*next)(double)) {
  double s = 1;
  for(double i = a; i < b + 1e-14; i = next(i)) {
    s = s * f(i);
  }

  return s;
}
```

Функции от по-висок ред

Да се напише функция `accumulate`

$$f(a)\ op\ f\big(next(a)\big)\ op\ \ldots\ op\ f(b)$$

Функции от по-висок ред

$$f(a)\ op\ f\big(next(a)\big)\ op\ \ldots\ op\ f(b)$$

```
double accumulate(double (*op)(double, double), double null_val,
  double a, double b,
  double (*f)(double), double (*next)(double)) {

  double result = null_val;

  for(double i = a; i < b + 1e-14; i = next(i)) {
    result = op(result, f(i));
  }

  return result;
}
```

Функции от по-висок ред

Оператор typedef

```
typedef <тип> <име>;
```

- `<име>` - идентификатор, определящ новото име на типа;
- `<тип>` - е дефиниция на тип;

Семантика:
Определя `<име>` за алтернативно име на `<тип>`.

Функции от по-висок ред

Пример:

```
typedef unsigned int positiveNumbers;

int main() {
  positiveNumbers a = 5;
  return 0;
}
```

Функции от по-висок ред

Пример:

```
double next(double a) {
  return a;
}

typedef double (*functionType)(double);

int main() {
  functionType func = next;
  return 0;
}
```

Функции от по-висок ред

$$f(a)\ op\ f\big(next(a)\big)\ op\ \ldots\ op\ f(b)$$

```
typedef double (*type1)(double, double);
typedef double (*type2)(double);

double accumulate(double (*op)(double, double), double null_val,
  double a, double b,
  double (*f)(double), double (*next)(double));


double accumulate(type1 op, double null_val,
  double a, double b,
  type2 f, type2 next);
```

Функции от по-висок ред

Функциите като оценка

```
typedef double (*functionType)(double);

functionType getOperation(char ch) {
  switch(ch) {
    case 'a': return sin;
    case 'b': return cos;
    case 'c': return exp;
    case 'd': return log;
  }
}


int main() {
  cout << getOperation('a')(0.5);
  return 0;
}
```

Следва продължение…