PDP
IT-ACADEMY

Function overloading.

Recursive functions

Reja:

- Function overloading
- Recursive functions
- Amaliy mashqlar

Function overloading

Dasturlashda nomlari bir xil, parametrlarining turi va soni bilan farq qiladigan funksiyalar **qayta yuklanuvchi funksiyalar** deyiladi.

PDP IT-ACADEMY

Example 1

```
float absolute(float var) {
    // code
}

int absolute(int var) {
    // code
}

int main() {

    absolute(-5);

    absolute(5.5f);

    ... ...

}
```

PDP IT-ACADEMY

```
// function with float type parameter
float absolute(float var){
    if (var < 0.0)
        var = -var;
    return var;
}

// function with int type parameter
int absolute(int var) {
     if (var < 0)
         var = -var;
    return var;
}
```

```
int main() {

    // call function with int type parameter
    cout << "Absolute value of -5 = " << absolute(-5) << endl;

    // call function with float type parameter
    cout << "Absolute value of 5.5 = " << absolute(5.5f) << endl;
    return 0;
}
```

PDP IT-ACADEMY

Output

```
Absolute value of -5 = 5
Absolute value of 5.5 = 5.5
```

Example 2

```
void display(int var1, double var2) {
    // code
}

void display(double var) {
    // code
}

void display(int var) {
    // code
}

int main() {
    int a = 5;
    double b = 5.5;

    display(a);

    display(b);

    display(a, b);

    ... ...

}
```

PDP IT-ACADEMY

```
// function with 2 parameters
void display(int var1, double var2) {
    cout << "Integer number: " << var1;
    cout << " and double number: " << var2 << endl;
}

// function with double type single parameter
void display(double var) {
    cout << "Double number: " << var << endl;
}

// function with int type single parameter
void display(int var) {
    cout << "Integer number: " << var << endl;
}
```

PDP IT-ACADEMY

```
int main() {

    int a = 5;
    double b = 5.5;

    // call function with int type parameter
    display(a);

    // call function with double type parameter
    display(b);

    // call function with 2 parameters
    display(a, b);

    return 0;
}
```

PDP IT-ACADEMY

Output

```
Integer number: 5
Float number: 5.5
Integer number: 5 and double number: 5.5
```

PDP
IT-ACADEMY

Recursive functions

Funksiya o'ziga o'zi to'g'ridan-to'g'ri yoki qandaydir vosita orqali murojaat qilish jarayoniga **rekursiya** deyiladi va bunday funksiya **rekursiv funksiya** deb ataladi.

PDP IT-ACADEMY

```
void recurse() {
    ... .. ...
    recurse();
    ... .. ...
}

int main() {
    ... .. ...
    recurse();
    ... .. ...
}
```

recursive call

function call

Har qanday to'g'ri tuzilgan rekursiya asosini ikkita shart tashkil qiladi:

- ✓ funksiyaning o'ziga o’zi murojaat qilishi;
- ✓ rekursiyaning to’xtash sharti.

PDP IT-ACADEMY

```
// Factorial of n = 1*2*3*...*n

int factorial(int n) {
    if (n > 1) {
        return n * factorial(n - 1);
    } else {
        return 1;
    }
}
```

PDP IT-ACADEMY

```
int main() {
    int n, result;

    cout << "Enter a non-negative number: ";
    cin >> n;

    result = factorial(n);
    cout << "Factorial of " << n << " = " << result;
    return 0;
}
```

PDP IT-ACADEMY

```
int main() {
    ... .. ...
    result = factorial(n);
    ... .. ...
}
```

n = 4

```
int factorial(int n) {
    if (n > 1)
        return n * factorial(n-1);
    else
        return 1;
}
```

n = 3

```
int factorial(int n) {
    if (n > 1)
        return n * factorial(n-1);
    else
        return 1;
}
```

n = 2

```
int factorial(int n) {
    if (n > 1)
        return n * factorial(n-1);
    else
        return 1;
}
```

n = 1

```
int factorial(int n) {
    if (n > 1)
        return n * factorial(n-1);
    else
        return 1;
}
```

4 * 6 = 24 is returned

3 * 2 = 6 is returned

2 * 1 = 2 is returned

1 is returned

PDP IT-ACADEMY

Output

```
Enter a non-negative number: 4
Factorial of 4 = 24
```

Amaliy mashqlar

Function overloading yordamida quyidahi **add()** funksiyasini yozing:

```
float add(float, float);
float add(float, float, float);
string add(string, string)
```

Parametr sifatida 2 ta son berilsa ham, 3 ta son berilsa ham ularning kattasini topib, qaytaruvchi **findMax()** funksiyasini "**function overloading**" yordamida yozing.

2 ta sonni parametr sifatida qabul qilganda ularning yig’indisini, 3 ta sonni parametr sifatida qabul qilganda esa ularning ko’paytmasini hisoblab, natija sifatida qaytaruvchi **calculate()** funksiyasini yozing.

1 dan N gacha bo’lgan natural sonlarning yig’indisini rekursiv funksiya yordamida hisoblang.

Berilgan N natural sonning raqamlari yig'indisini rekursiv funksiya yordamida hisoblang.

n-Fibonachi sonini rekursiv funksiya yordamida aniqlovchi dastur tuzing.

Bunda:

$f(0) = 1$, $f(1) = 1$, $f(n) = f(n-1)+f(n-2)$, $n>=2$.

Kompyuter tomonidan 1 dan 10 gacha oraliqdagi tasodifiy natural son generatsiya qilinadi. Ushbu tasodifiy sonni foydalanuvchi tomonidan kiritilgan son bilan solishtiruvchi rekursiv funksiya yozing va bu funksiya foydalanuvchi tasodifiy sonni to’g’ri topmagunicha rekursiv tarzda chaqirilsin.

PDP
IT-ACADEMY

E`tiboringiz uchun rahmat!