



Кафедральный практикум

5 семестр

Часть 1. Введение в функциональное программирование (продолжение)

<http://sp.cmc.msu.ru/~kornyxin/fp/slides/part1-2.pdf>



План

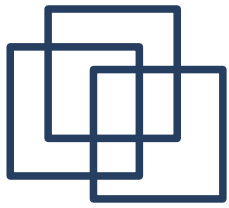
Часть 1.

- 1) Организационные вопросы
- 2) **Функциональный стиль программирования**

Часть 2. Язык программирования Scheme.

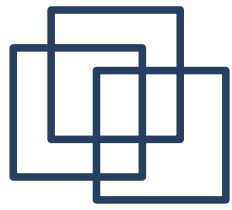
Часть 3. Функции высшего порядка.
«Векторное» мышление.

Часть 4. Теоретический фундамент ФП.



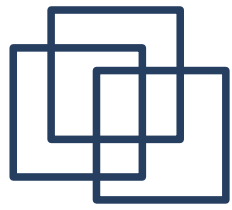
Орг.вопросы

- Что предстоит сдавать:
 - *Домашние работы (ejudge)*
 - *Практические задания (виртуальная машина и генетическое программирование)*
 - *Контрольные работы в классе*
- Материалы: <http://sp.cmc.msu.ru/~kornukhin>
- Вопросы, пожелания, предложения:
 - kornevgen@cmc.msu.ru (Корныхин Евгений)
 - amonakov@ispras.ru (Монаков Александр)



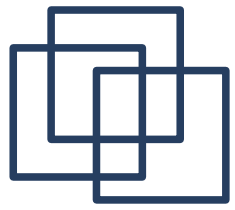
«Expression only» стиль

- примеры statement'ов :
 - присваивание (всех видов, в том числе ++)
 - if (...) { ... } else { ... }
 - while (...) { ... }
 - for (...; ...; ...) { ... }
 - «expression statement» (н-р, «x+1;»)
 - return ...;
- примеры expression'в :
 - арифметика, булевские выражения и т.п.
 - вызов функции
 - оператор ? :



«Expression only» стиль

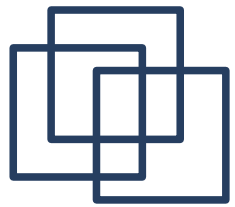
- «Expression only» - значит, «без statement'ов», за исключением return.
- В простейшем случае, тело функции — return с выражением.
- Разрешаем использовать константные локальные переменные.
- Разрешаем использовать «if-then-else», но при условии, что он ведет себя как «? :».



Проверка на простоту

```
int is_prime2(const int n, const int d);
int is_prime(const int n)
{
    return is_prime2(n, n/2);
}
```

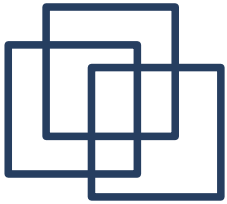
```
int is_prime2(const int n, const int d)
{
    if (d == 1) return 1;
    else if ((n % d) == 0) return 0;
    else return is_prime2(n, d-1);
}
```



Проверка на простоту

```
list<int> divisors(const int n);
int count_of(const list<int> ls);

int is_prime(const int n)
{
    return (n > 1) &&
           count_of(divisors(n)) == 2;
}
list<int> divisors(const int n)
{
    return filter(range(1, n), |i|(n%i == 0));
}
```



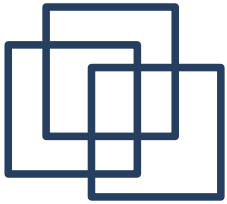
На C++

- Стил «expression only» на C++ (без if):
 - все формальные параметры функций — константные
 - отсутствие глобальных переменных
 - все локальные переменные — константные
 - нет функций, возвращающих void
 - нет statement'ов, за исключением return
 - только один return в функции
 - всё остальное отследит компилятор
- Доп. опции компилятора g++: -Wall -Werr



На C++

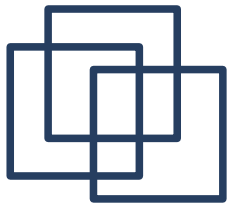
- Стиль «expression only» на C++ (**c if**)
дополнительно:
 - нет statement'ов за исключением return **и if-then-else**
 - ~~только один return в функции~~
 - if обязательно содержит else
 - все ветви каждого if заканчиваются return
 - нет «мёртвого» кода
 - всё остальное отследит компилятор
- Доп. опции компилятора g++: -Wall -Werr



Задача

```
#include "fplists.h"
int max(const list xs)
{
    // xs is list of natural numbers
    // if xs is empty then return 0
    // else return maximum number from xs
}
```

```
int len(const list xs); - длина
int first(const list xs); - первый эл-т
int second(const list xs); - второй эл-т
list tail(const list xs); - «ХВОСТ»
```



Задача: решение

```
#include "fplists.h"
int max(const list xs)
{
    if (len(xs) == 0) return 0;
    else if (len(xs) == 1) return first(xs);
    else {
        const int m = max(tail(xs));
        const int f = first(xs);
        if (f > m) return f;
        else return m;
    }
}
```

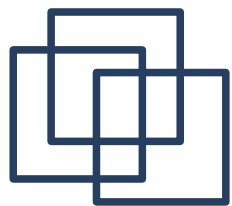


Задача: еще решение

```
#include "fplists.h"
int max(const list xs)
{
    return max_tail(xs, 0);
}

int max_tail(const list xs, const int cmax)
{
    if (len(xs) == 0) return cmax;
    else return max_tail(tail(xs),
                        max(first(xs), cmax) );
}
```

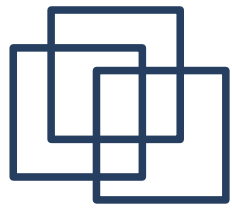
ЧЕМ ЭТО РЕШЕНИЕ ЛУЧШЕ ПРЕДЫДУЩЕГО ?



Задача: еще решение

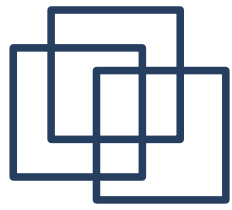
```
#include "fplists.h"
int max(const list xs)
{
    return foldr(|i,j|(i>j?i:j), 0, xs);
}
```

ЧЕМ ЭТО РЕШЕНИЕ ЛУЧШЕ ПРЕДЫДУЩИХ ?



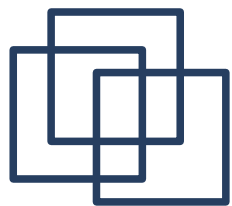
«Expression only» стиль

- Тело_функции ::=
{
 {Объявление_лок_перем}
 (**return** выражение ;
 | **if** (выражение) Тело_функции
 else Тело_функции
)
}
- Объявление_лок_перем ::=
 const Тип Имя = выражение ;



«Expression only» стиль

- Тело_функции ::= Возврат | Блок
- Возврат ::= **return** выражение ;
| **if** (выражение) Тело_функции
else Тело_функции
- Блок ::=
{
 {Объявление_лок_перем}
 Возврат
}
- Объявление_лок_перем ::=
const Тип Имя = выражение ;



Работа с «кучей»

- Все указатели должны быть **const T * const**
- Нужен struct → пишем в нем конструктор, в котором инициализируем поля структуры.
- Выделение памяти при помощи **new**.
- Для списков можно использовать fplists

```
const int * const inc(const int * const t)
{
    return new int(*t + 1);
}
```




fplists

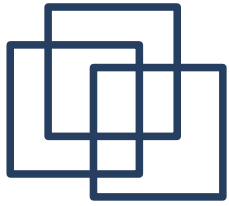
```
struct _list {
    int value;
    const struct _list *
const next;
    _list(int v, const
struct _list *const n) :
value(v), next(n) {}
};

typedef const struct _list
* const list;

int len(const list xs);
int first(const list xs);
int second(const list xs);
list tail(const list xs);

list empty(void)
{ return 0; }

list create(const int
value, const list rest)
{ return new _list(value,
rest); }
```



Создание СПИСКОВ

- Выражение, результатом которого является список из элементов 1, 2, 3 :

```
create(1, create(2, create(3, empty())))
```



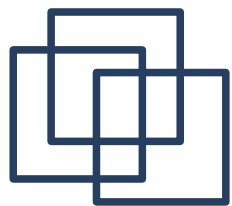
Задачи

- Попробуйте реализовать в стиле «expression only» следующие функции:
 - 1) дан непустой список, вернуть его последний элемент
 - 2) дано число, вернуть список всех его простых делителей без повторов (порядок не важен)
 - 3) даны два списка, вернуть их конкатенацию



«Хвостовая» рекурсия

- Все `return` в каждой рекурсивной функции `f` имеют вид:
 - `return` выражение без функций;
 - ИЛИ `return f(...);`
- Рекурсивные вызовы функции не встречаются в `if-then-else` и при объявлении локальных переменных
- Попробовать решить некоторые прошлые задачи (и прошлые домашние задачи), применяя «хвостовую» рекурсию
- Чем она хороша ?



Домашнее задание

- 1) (не сделали в прошлый раз? сделаем его сейчас!)
Какую программу/задачу все знают?
- 2) Реализовать в стиле «из одних выражений»
несколько функций на C++ с использованием fplists
 - нерешенные задачи с семинара
 - <http://sp.cmc.msu.ru/~kornyxhin> → hw #2.1
- 3) Решения оформлять в виде текстовых файлов и
присылать мне на почту kornevgen@cmc.msu.ru
- 4) Подумать, что дает такой стиль написания
программ для качества программ, дает ли новую
парадигму программирования