

ОСНОВЫ

# НЕОБХОДИМЫЙ МИНИМУМ

- Прекрасное владение языком программирования
- Уверенное знание большого количества алгоритмов (уверенно знать – значит уметь быстро, без подготовки реализовать алгоритм)
- Математическая подготовка
- Большое количество прорешенных задач
- Опыт участия в тренировочных и реальных олимпиадах
- Психологическая подготовка

# ЛИТЕРАТУРА

- Окулов С.М. «Программирование в алгоритмах», 2004
- Порублев И.Н., Ставройский А.Б. «Алгоритмы и программы. Решение олимпиадных задач», 2007
- Меньшиков Ф.В. «Олимпиадные задачи по программированию», 2006
- Андреева Е.В. «Математические основы информатики. Элективный курс», 2012
- Шень А. «Программирование. Теоремы и задачи», 2004

# САЙТЫ

- Учебные курсы [www.intuit.ru](http://www.intuit.ru)
- Коллекция алгоритмов <http://e-maxx.ru/algo>
- Международные и всероссийские олимпиады по информатике  
<http://info.rusolymp.ru>
- Сайт школьных олимпиад, проводимых в Приморском крае  
<http://imcs.dvgu.ru/works/school.html>
- Площадка соревнований по программированию <http://codeforces.ru/>
- Дистанционная подготовка школьников по информатике  
<http://informatics.mscme.ru>
- Сайт «Школа программиста» Красноярского края <http://acmp.ru>

# ЯЗЫКИ

Конкретный язык не существенен – задачи спортивного программирования успешно решаются на любых языках программирования (возможно, за исключением эзотерических).

В рамках занятий задачи будут рассматриваться на языке C++

# ПРАВИЛА ХОРОШЕГО ТОНА

- Не забывайте о проектировании программ сверху вниз:
  - прежде чем приступить к кодированию, вы должны спроектировать программу на достаточном уровне детализации на бумаге.
  - Не приступайте к кодированию до тех пор, пока не сможете ясно, понятно для любого слушателя рассказать идею решения
- Разбивайте программу на отдельные подпрограммы (процедуры и функции). Старайтесь отлаживать каждую функцию по отдельности

# ПРАВИЛА ХОРОШЕГО ТОНА

- Старайтесь использовать как можно меньше глобальных переменных: процедуры и функции должны быть максимально независимыми
- Всегда программируйте «с отступами»
- Выбирайте осмысленные имена для переменных, функций и т.д.
- Одна строка – один оператор
- Не забывайте присваивать переменным начальные значения, даже если компилятор сделает это за вас
- Добавляйте комментарии по ходу написания программы
- Не пренебрегайте тестированием программы. Помните, что каждая последняя ошибка – есть предпоследняя.

# ОСОБЕННОСТИ РЕШЕНИЯ ОЛИМПИАДНЫХ ЗАДАЧ

- Программа представляет собой консольное приложение
- Как правило, исходные данные должны считываться из исходного файла и записываться в выходной файл. Все файлы текстовые.
- **Проверять корректность данных в исходном файле не требуется!**
- **Необходимо тщательно следить за корректностью данных, которые записываются в выходной файл.**

# ОСОБЕННОСТИ РЕШЕНИЯ ОЛИМПИАДНЫХ ЗАДАЧ

- Заданы ограничения на время и на ресурсы памяти (программа должна выполняться не дольше предъявленного лимита и не превышать требований к допустимому объему памяти), т.е. необходимо работать над эффективностью программы
- Решения проверяются автоматизированной системой по заранее заготовленному большому набору тестов (порядка 30 – 40).

# ТИПЫ ДАННЫХ C++

ТИП	ПАМЯТЬ	ДИАПАЗОН
bool	1	0 / 255
char	1	0 / 255
short int	2	-32 768 / 32 767
unsigned short int	2	0 / 65 535
int	4	-2 147 483 648 / 2 147 483 647
unsigned int	4	0 / 4 294 967 295
long int	4	-2 147 483 648 / 2 147 483 647
unsigned long int	4	0 / 4 294 967 295

# ТИПЫ ДАННЫХ C++

тип	память	диапазон
<b>float</b>	<b>4</b>	<b>-2 147 483 648.0 / 2 147 483 647.0</b>
<b>long float</b>	<b>8</b>	<b>-9 223 372 036 854 775 808 .0 / 9 223 372 036 854 775 807.0</b>
<b>double</b>	<b>8</b>	<b>-9 223 372 036 854 775 808 .0 / 9 223 372 036 854 775 807.0</b>

# ВВОД/ВЫВОД В КОНСОЛЬ

Из языка C:

printf

scanf

Из языка C++:

std::cout<<

std::cin>>

# УСЛОВИЯ

if (условие)

{

else }

switch (парам)

{

case a:

...

break;

case b:

...

break;

default:

....

break;

}

b=условие?x:y;

# ЦИКЛЫ

for – цикл со счетчиком

while – цикл с условием

do-while – цикл с постусловием

```
int a=0;
while (a==1) //ни разу не выполнится
    cout<<a;
do{ //выполнится один раз
cout<<a;
}
while (a==1);
```

# МАССИВЫ

int \*a=new int[n]; //динамический – размер можно задать по ходу программы

int b[10]; //статический

int c[10][5]; //статический двумерный

int \*\* d=new int\*[n]; //динамический двумерный

for (int i=0;i<n;i++)

    d[i]=new int[m];

# ОПЕРАЦИИ

	a	b	a   b	a & b	a ^ b	a >> 1	a << 1	~a
+ – сложение	0	0	0	0	0	1	0	1
- – вычитание	0	1	1	0	1	1	0	1
* – умножение	1	0	1	0	1	0	10	0
/ – деление	1	1	1	1	0	0	10	0

% – остаток от деления

^ – XOR – исключающее ИЛИ

& – И

| – ИЛИ

>> – битовый сдвиг вправо

<< – битовый сдвиг влево

~ – инверсия битов

# ФУНКЦИИ

```
int func (int a, int b){  
    return a+b;  
}
```

Функции описываются по порядку их следования в программе.

```
int f1 (int a, int b) { //Не заработает  
    int c=f2(a,b);    // f2 тут ещё не объявлена  
    return c+a+b;  
}  
int f2 (int a, int b){  
    return a*b;  
}
```

# ФУНКЦИИ

```
int f2 (int a, int b);
int f1 (int a, int b) { //заработает
    int c=f2(a,b);
    return c+a+b;
}
...
int f2 (int a, int b){
    return a*b;
}
```

Прототип – объявление функции с отложенным описанием

# ФАЙЛЫ

```
#include <fstream>

ifstream in; //поток для чтения

ofstream out; //поток для записи

string d;

in.open("file.txt");

out=new ofstream ("file2.txt");

in>>d;

ofstream<<d<<endl;
```

# СТРОКИ

```
#include <string>
```

```
string s="asdadasdasd";
```

Операции:

Конкатенация (сложение строк)

Поиск вхождения

Выделение подстроки

И т.д.