

ПРОГРАММИРОВАНИЕ (часть 2)

ОСНОВНЫЕ ВОПРОСЫ ДЛЯ ПРОРАБОТКИ

1. Структурирование программ
2. Прототипы функций и стиль их описания.
3. Классы памяти и области видимости переменных в функциях.
4. Типы возвращаемых значений.
5. Способы передачи аргументов в функции.
6. Передача массивов в качестве аргументов
7. Аргументы по умолчанию.
8. Полиморфизм (перегрузка) функций.
9. Структуры и классы в C++.
10. Базовые концепции классов.
11. Инкапсуляция данных и методов.
12. Объекты, указатели на объекты и способы доступа к ним.
13. Конструкторы и деструкторы.
14. Файлы: потоковый ввод-вывод.
15. Открытие/закрытие файлов.
16. Запись в файл.
17. Чтение из файла.
18. Прямой доступ при использовании файлов.
19. Простой вид списков – однонаправленный линейный, использование указателей.
20. Базовые концепции однонаправленного линейного списка: элемент списка (описание с использованием структуры/класса), указатель на следующий элемент.
21. Базовые операции с однонаправленным линейным списком: создание пустого списка, удаление всего списка.
22. Простой вид списков – однонаправленный линейный, использование формуляра указателей.
23. Основные операции с однонаправленным линейным списком: вставка элемента (варианты).
24. Основные операции с однонаправленным линейным списком: удаление элемента (варианты).
25. Хранение и обработка текстовой информации: представление строки с маркером.
26. Хранение и обработка текстовой информации: использование списка для работы с набором строк.
27. Факультативно: библиотека обработка символьных строк.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАЧИ

----- 1 задача из раздела -----

Задача A32. Все ли вызовы функции MyFunc допустимы

```
void MyFunc (int a,int b=1,int c=2);
```

```
//.....
```

```
MyFunc(x,y,z); MyFunc(x,y); MyFunc(x); MyFunc( );
```

Задача A33. Все ли вызовы функции MyFunc допустимы

```
void MyFunc (int a,int b,int c=0);
```

```
//.....
```

```
MyFunc(x,y,z); MyFunc(x,y); MyFunc(x); MyFunc( );
```

Задача A34. Все ли вызовы функции MyFunc допустимы

```
void MyFunc (int a=0,int b=0,int c=0);
```

```
//.....
```

```
MyFunc(x,y,z); MyFunc(x,y); MyFunc(x); MyFunc( );
```

Задача A41. В каком месте приведенного фрагмента содержится ошибка:

```
void f(int a) {return a*a;} //1
```

```
void main()
```

```
{ int a=2; //2
```

```
cout<<f(a); //3
```

```
}
```

Задача A42. В каком месте приведенного фрагмента содержится ошибка:

```
int f(int a) {return a*a;} // 1
void main()
{ int x,y; //2
  cout<<f(x); //3
}
```

----- 2 задачи из раздела -----

Задача А35. Каков результат работы приведенного фрагмента

```
void f (int a)
{cout<<"Функция1";}
void f (float b)
{cout<<"Функция2";}
void main()
{int x=1;
 float y=2;
 f(y);
}
```

Задача А43. Каков результат работы приведенного фрагмента

```
int f(int a,int b=2){return a+b;}
void main()
{ int x=1;
  cout<<f(x)+3;
}
```

Задача А44. Каков результат работы приведенного фрагмента

```
int f(int a,int b=2){return a+b;}
void main()
{ int x=1,y=1;
  cout<<f(x,y)+3;
}
```

Задача А45. Каков результат работы приведенного фрагмента

```
int f(int a) {return a;}
void main()
{int x=2;
  cout<<f(x)+1;
}
```

Задача А46. Каков результат работы приведенного фрагмента

```
float f(int a) {return a*a;}
void main()
{int x=3;
  cout<<f(x);
}
```

Задача А47. Каков результат работы приведенного фрагмента

```
char f(char a) {return a;}
void main()
{char ch='A';
  cout<<f(ch);
}
```

Задача А48. Каков результат работы приведенного фрагмента

```
void f(char a) {cout<<a;}
void main()
{char ch='A';
  f(ch);
}
```

Задача А49. Каков результат работы приведенного фрагмента

```
float a[5]={0};
for(int i=1;i<5;i++)
```

```
cout<<a[i]<<" ";
```

----- 2 задачи из раздела -----

Задача В4-1. Каков результат работы программы

```
int f ( );
void main ( )
{ cout<<f ( )<<" "; cout<<f ( ); }
int f ( )
{ static int a = 0;
  for (int k =1; k<=3; k++) a+=k;
  return a; }
```

Задача В4-2. Каков результат работы программы

```
int i = 1;
int a( );
void main ( )
{cout<<i<<" "; a ( ); cout<<i++; }
int a ( )
{int i =5;
  cout<<i++<<" "; }
```

Задача В4-3. Каков результат работы приведенного фрагмента

```
void f (int a)
{cout<<"Вызвана Функция a";}
void f (float b)
{cout<<"Вызвана функция b";}
void main()
{int x=1;
 float y=2.6;
 f(y);
 f(x);
}
```

Задача В4-4. Каков результат работы приведенного фрагмента

```
void f (char a)
{cout<<"1000";}
void f (float b)
{cout<<"2000";}
void main()
{char x='B';
 float y=1.0;
 f(x);
 f(y);
}
```

----- 1 задача из раздела -----

Задача В4-6. Каков результат работы приведенного фрагмента

```
float f(float a,float b) {return a+b;}
void main()
{float x=2.3, y=2;
 cout<<f(x,y);
}
```

Задача В4-7. Каков результат работы приведенного фрагмента

```
float f(float a,float b) {return a*b;}
void main()
{float x=2.3, y=0;
 cout<<f(x,y);
}
```

Задача В4-8. Каков результат работы приведенного фрагмента

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

```
float f(float a,int b) { return a*b;}
void main()
{ float x=3.0;
  int y=2;
  cout<<f(x,y)<<endl;
  cout<<f(y,x)<<endl;
}
```

----- 1 задача из раздела -----

Задача B5-4. Какой оператор приведенного фрагмента содержит ошибку

```
class X
{int i,j;//1
 float a; //2
};
void main()
{ X ob; //3

  ob.i=1; //4
}
```

Задача B5-5. Какой оператор приведенного фрагмента содержит ошибку

```
class X
{int i,j;//1
 float s(); //2
};
void main()
{ X ob; //3
  cout<<ob.s(); //4
}
```

----- 1 задача из раздела -----

Задача B5-1. Каков результат работы приведенного фрагмента

```
class X
{public:
  int a,b;
  X(int x, int y)
  { a=x; b=y; cout<<"a="<<x<<"b="<<y;}
};
void main()
{ X ob(1,2);
}
```

Задача B5-2. Каков результат работы приведенного фрагмента

```
class MyClass
{public:
  char c1,c2;
  MyClass(char x, char y)
  { c1=x; c2=y; cout<<"c1="<<x<<" c2="<<y;}
};
void main()
{ MyClass mc('A','B');
}
```

Задача B5-3. Каков результат работы приведенного фрагмента

```
class complex
{public:
  double re,im;
  complex(double x, double y)
  { re=x; im=y; cout<<"re="<<x<<" im="<<y;}
};
```

```
void main()
{ complex z(1.0,2.0);
}
```

Задача B5-6. Каков результат работы приведенного фрагмента

```
class complex
{double re,im;
public:
complex(double x, double y=0)
{re=x; im=y; cout<<"re="<<x<<" im="<<y;}
};
```

```
void main()
{ complex z(10.0,2.0);
}
```

Задача B5-7. Каков результат работы приведенного фрагмента

```
class complex
{double re,im;
public:
complex(double x, double y=0)
{re=x; im=y; cout<<"re="<<x<<" im="<<y<<endl;}
};
```

```
void main()
{ complex z1(1.0), z2(1.0,2.0);
}
```

Задача B5-8. Каков результат работы приведенного фрагмента

```
class complex
{public:
double re,im;
complex(double x, double y)
{re=x; im=y;}
};
void main()
{ complex z(1.0,20.0);
cout<<"re="<<z.re<<" im="<<z.im;
}
```

Задача B5-9. Каков результат работы приведенного фрагмента

```
class complex
{public:
double re,im;
complex(double x, double y)
{re=x; im=y;}
};
void main()
{ complex *z=new complex(3.0,2.0);
cout<<z->re<<" "<<z->im;
}
```

Задача B5-10. Каков результат работы приведенного фрагмента

```
class Rect
{int a,b;
public:
Rect(int x, int y) {a=x; b=y;}
int s(){ return a*b;}
};
void main()
{ Rect rct(4.0,3.0);
cout<<rct.s();
```

}

Задача B5-11. Каков результат работы приведенного фрагмента

```
class Rect
{int a,b;
public:
Rect(int x, int y) {a=x; b=y;}
int s(){ return a*b;}
};
void main()
{ Rect*rct=new Rect(2.0,3.0);
cout<<rct->s();
}
```

----- 1 задача из раздела -----

Задача C1-1.

Отличие потоков ifstream, ofstream и fstream для работы с файлами.

Задача C1-2.

```
При обращении к функции readFile:      #include <fstream.h>
char readFile(const char *fileName)
{char s; ifstream inputFile( fileName );
  if( !inputFile ) s='1'; else s='0';
  return s;
}
```

был возвращен символ 1. Что это означает?

Задача C1-3.

Что означает режим ios::app, указываемый при открытии файла.

Задача C1-4.

Что позволяет выполнить следующая функция (приводится начало ее тела):

```
#include <fstream.h>
void Files(const char *fileName)
{fstream inputFile;
  inputFile.open(fileName,ios::out);
  ...
}
```

Задача C1-5.

Какое действие обеспечивает следующий фрагмент программы:

```
char s;      fstream inputFile;
while(!inputFile.eof())
{ s=inputFile.get();
  if(!inputFile.eof()) cin<<s;
}
```

----- 1 задача из раздела -----

Описание элемента списка имеет вид:

```
struct list { int val; list *next; };
```

Определите вид списка, «смысл» каждого указателя, выполняемое действие над списком:

Задача D1-1.

```
int F1(list *p)
{ int n;
for (n=0; p!=NULL; p=p->next, n++);
return n;
}
```

Задача D1-2.

```
list* F2(list *ph, int v)
{ list *q = new list;
q->val = v; q->next = ph; ph = q;
return ph;
}
```

```
}
```

Задача D1-3.

```
list* F3(list *p, int n)
{ for (; n!=0 && p!=NULL; n--, p=p->next);
return p;
}
```

Задача D1-4.

```
list* F4(list *ph, int v)
{ list *p,*q = new list;
q->val = v; q->next = NULL;
if (ph == NULL) return q;
for ( p=ph ; p->next !=NULL; p = p->next);
p->next = q; return ph;
}
```

Задача D1-5.

```
list* F5(list *ph, int n)
{ list *q ,*pr,*p;
for ( p=ph,pr=NULL; n!=0 && p!=NULL; n--, pr=p, p =p->next);
if (p==NULL) return ph;
if (pr==NULL) { q=ph; ph=ph->next; }
else { q=p; pr->next=p->next; }
delete q;
return ph;
}
```

Задача D1-6.

```
int F6(list *p)
{ int n; list *q;
if (p==NULL) return 0;
for (q = p, p = p->next, n=1; p !=q; p=p->next, n++);
return n;
}
```

Задача D1-7.

```
list* F7(list *ph)
{ list *q, *out, *p , *pr;
out = NULL;
while (ph !=NULL)
{ q = ph; ph = ph->next;
for ( p=out,pr=NULL; p!=NULL && q->val>p->val; pr=p,p=p->next);
if (pr==NULL)
{ q->next=out; out=q; }
else
{ q->next=p; pr->next=q; }
} return out;
}
```

Задача D1-8.

```
list* F10(list *ph, int v)
{ list *q ,*pr,*p;
q=new list; q->val=v; q->next=NULL;
if (ph==NULL) return q;
for ( p=ph,pr=NULL; p!=NULL && v>p->val; pr=p,p=p->next);
if (pr==NULL) { q->next=ph; ph=q; }
else { q->next=p; pr->next=q; }
return ph; }
```

ЛИТЕРАТУРА

1. С.Поттс, Т.С.Монк. Borland C++ в примерах, 1996.
2. У.Мюррей, К.Паппас. Visual C++ , 1996.
3. П.Кимел. Borland C++, 1997.
4. У.Севитч. C++ в примерах, 1997.
5. В.И.Анисимов. Программирование, СПбГЭТУ, 2005.
6. Б.Страуструп. Язык программирования C++, 1991 - 1998.
7. Страуструп Б. Язык программирования C++. Специальное издание. – СПб.: Бином, Невский Диалект, 2008. – 1104 с.
8. Дейтел Х. М., Дейтел П. Дж. Как программировать на C++. – М.: Бином-Пресс, 2008. – 1456 с. (2009. – 800 с.)
9. Д.Дж. Круглински, С.Уингоу, Дж. Шеферд. Программирование на Microsoft Visual C++ 6.0 для профессионалов: [учебное пособие] / Пер. с англ. - СПб.: Питер, 2001. - 819 с.
10. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989. – 360 с.
11. Калмычков В. А. Использование структурированных данных при программировании на языке C++: Учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2011.