



Экзамен по курсу «Языки программирования»

Вариант 1. (1999)

См. оригинал [здесь](#) .

1. Дайте определение интерфейса в языке Ява. Как интерфейсы могут быть реализованы на языке C++?
2. Напишите спецификацию шаблонной функции на языке Си++, эквивалентной спецификации родовой процедуры на языке Ада, приведенной ниже.

```
generic
  type ELEM is private;
  type INDEX is (<>);
  type VECTOR is array (INDEX) of ELEM;
  with function «>» (A,B: in ELEM) return BOOLEAN is <>;
procedure G_SORT (A: in out VECTOR);
```

3. Как реализована динамическая проверка типа в языке Оберон (перечислите все средства)?
4. Напишите (и объясните ответ), что будет выдано в стандартный канал вывода при выполнении вызова функции Q в следующем фрагменте программы:

```
int i = 0;
class STRING {
  char * cp;
public :
  STRING (const char * s){
    cp = new char [strlen (s) + 1];
    strcpy (cp, s);
```

```

}
STRING (const STRING& s){
    cp = new char [strlen (s.cp) + 1];
    strcpy (cp, s.cp);
}
STRING (const STRING& s1,const STRING& s2){
    cp = new char [
        strlen(s1.cp)+strlen (s2.cp) + 1
    ];
    strcat (strcpy (cp, s1.cp), s2.cp);
}
~STRING () {
    i++; delete [] cp;
}
operator char * () { return cp; };
void out () { cout << cp << '\n'; }
};
void f(STRING s) { s.out();}
void q()
{ STRING p ("string1");
  STRING s("string2","string3");
  f (p); f (s);
}
void Q(){ q(); cout << i; }

```

5. Напишите фрагмент программы на языке Си++, эквивалентный фрагменту программы на языке системы Delphi.

```

try
    G(1);
finally
    i := 5;
end;

```

6. Дайте определение и пример виртуального множественного наследования в языке Си++.
7. Перечислите способы передачи параметров в языках программирования. Какие из этих способов реализованы в языках Си и Паскаль?
8. Что будет выдано в стандартный канал вывода при вызове функции $F()$? Преобразуйте спецификацию классов так,

чтобы было выдано:

```
1 1 2 2
2 2 3 3
3 3 3 3
```

Единственным допустимым видом преобразования является добавление спецификатора *virtual*.

```
class X {
public:
    void g() { cout << 1 << ' '; }
    void f() { g(); }
};

class Y: public X {
public:
    void g() { cout << 2 << ' '; }
    void f() { g(); }
};

class Z: public Y {
public:
    void g() { cout << 3 << ' '; }
    void f() { g(); }
};

X x; Y y; Z z;
X * px = &x; Y * py = &y; Z * pz = &z;

void out(void) {
    px->f(); px->g(); py->f(); py->g();
    cout << '\n';
}

void F(void) {
```

```
    out(); px = py; py = pz;  
    out(); px = pz; py = pz;  
    out();  
}
```