1. （单选）int a = 0x10; 请问以上声明是什么意思

A. a是十进制的10

B. a是八进制的10

C. a是二进制的10

D. a是十进制的16，16进制的10

2. （单选）以下代码的输出是什么？

#include <iostream>

#include <string>

void Alpha(string beta) {

    cout << beta << endl;

}

int main()

{

    Alpha("gamma");

    return 0;

{

A.编译错误

B.g

C.gamma

D.无输出

3.（单选）以下两段从代码属于两个不同的程序，在每个程序中，当proc被多线程并发调用了10000次后，对于相应的全局变量的值，说法正确的是（）

std::atomic<std::uint32\_t> g\_a(0);

void proc()

{

++g\_a;

}

volatile std::uint32\_t g\_v(0);

void proc()

{

++g\_v;

}

A、 g\_a的值为10000

g\_v的值不确定

B、 g\_a的值不确定

g\_v的值为10000

C、 g\_a的值为10000

g\_v的值为10000

D、 g\_a的值不确定

g\_v的值不确定

4. （多选）对于string类型的字符串mystring，哪些可以获取mystring所表示的字符长度（）

A、sizeof(mystring)

B、sizeof(mystring)-1

C、mystring.size()

D、mystrong.capacity()

E．mystring.length()

5.（多选）Data 是一个由32个unit32\_t组成的数据结构，DataList类的所有成员函数均可被多个线程调用，正确的是（）

class DataList {

public:

void AddToList(const Data &data);

void DelFronList();

Data &GetListHead();

int popListTail(Data &data ToFind);

private:

std::list<Data> m\_list;

std::mutex m\_mutex;

};

A、

void DataList::AddTolist(const Data &data)

{

std::lock\_guard<std::mutex> guard(m\_mutex);

m\_list.push\_back(data);

}

B、

void DataList::DelFormList()

{

std::lock\_guard<std::mutex> guard(m\_mutex);

if (!m\_list.empty()) {

m\_list.push\_back(data);

}

}

C、

Data &DataList::GetListHead()

{

if (m\_list.size() > 0) {

return m\_list.front();

}

static Data data;

return data;

}

D、

int DataList::poplistTail(Data &data)

{

if (!m\_list.empty()) {

std::loch\_guard<std::mutex> guard(m\_mutex);

data = m\_list.back();

m\_list.pop\_banck();

return 0;

}

return -1;

}

6. \*\*\*（多选）using typedef数组指针，指向数组的指针是

A. typedef int (Array\*)[10]

B. using Array = int(\*)[10]

C. typedef int (\*Array)[10]

D. using \*Array = int[10]

7.（单选）有一个main.c文件，共100行代码，假定任意一行都可以设置断点，其中有int main()和void Func(int a, int b)两个函数，编译后的文件执行发生死锁，需要分析相关的信息找到死锁相关代码，以下关于gdb获取死锁相关信息描述正确的是

A. 执行info address可以显示死锁的代码位置

B. 执行info threads显示所有线程信息，找到死锁相关线程

C. 执行info breakpoints显示断点，断点即死锁信息

D. 死锁时操作系统把进程挂起了，无法通过gdb获取任何信息

8. \*（单选）请问下面这段程序的输出结果是

#include <iostream>

#include <string>

using namespace std;

class Base {

public:

virtual ~Base(){}

virtual void Print() { std::cout<<"Base"<< endl;}

};

class Derived : public Base {

public:

virtual void Print() { std::cout<<"Derived"<< endl; }

};

int main()

{

Base\* base = new Derived();

base->Print();

base -> Base::Print();

delete base;

return 0;

}

A. Base

Base

B. Base

Derived

C. Derived

Base

D. Derived

Derived

9. （单选）以下选项中不符合编程规范的是

A. int errorCode = 5;

B. int hour = time \* 60 \* 60;

C. float mid = len / 2;

D. constexpr int SYSTEM\_VERSION = 4096;

10. \*\*（单选）以下代码执行完成后i j的值是

int main()

{

int i = 0;

int j = 0’

for (i = 10; i >= 0; --i) {

for (j = 10; j == 1; --j) {

i = j;

}

}

}

A. 0 0

B. -1 10

C. 执行超时

D. -1 0

11. \*\*（多选）以下#define定义正确的是

A. #ifdef \_\_GUN\_\_

#if (\_\_GUN\_\_ == 3)

…

#endif

#endif

B. #if defined(\_\_GUN\_\_) && \_\_GUN\_\_ > 3

…

#endif

C. #if (\_\_GUN\_\_ == 3)

…

#endif

D. #if \_\_GUN\_\_ == 3

…

#endif

12.（单选）GDB直接按下回车的作用是（）

A、next

B、step

C、continue

D、重复执行上一次命令

13. \*\*\*\*（单选）以下关于devid描述正确的是

typedef int DevId;

typedef DevId\* DevIdPtr;

const DevIdPtr devid;

A. 指向const DevId的指针，指针指向的内容不能被改变

B. 指向const DevId的指针，指针本身不能被改变

C. 指向DevId的const指针，指针本身不能被改变

D. 指向const DevId的const指针，指针本身合指针指向的内容都不能被改变

14. （单选）请问以下代码输出什么？

#include<iostream>

using namespace std;

int main()

{

int a = 1;

const int \*p=&a ;

\*p=9;

cout << \*p << endl;

return 0;

}

1. 编译错误
2. 9
3. 1
4. 0

15. \*\*\*（单选）以下为3个代码文件中的代码片段

// service.h

strcut Service {

static const std::string NAME;

};

// service.cpp

#include “service.h”

const std::string Service::NAME=“SERVICE1”;

// main.cpp

#include “service.h”

#include <iostream>

const std::string MSG=“service” + Service::NAME + “ready”;

int main()

{

std::cout<<MSG<<std::endl;

}

对于程序运行结果，以下说法正确的是

A. 运行输出 service SERVICE1 ready

B. 运行输出 service

C. 运行输出中包含随机字符或乱码

D. 运行结果取决于编译链接过程及std::string的内部实现

16. （多选）（考点完全相同的一个题）如下对于gtest字符串检查，描述正确的有

A. EXPECT\_STRNE(str1, str2)：两个字符串内容不同，检查通过

B. EXPECT\_STRNE(str1, str2)：两个字符串内容相同，检查通过

C. ASSERT\_GT(val1, val2)：val1的值大于val2时，断言失败

D. ASSERT\_GT(val1, val2)：val2的值大于val1时，断言失败

17.（单选）（考点完全相同的一个题）以下代码输出为

int main()

{

int x = 1;

int y = 2;

if (++x >= 2 || y++ >= 2) { }

std::cout<<x<<“ ”<<y<<std::endl;

return 0;

}

A. 2 2

B. 2 3

C. 3 3

D. 1 2

18.（单选）请问下面这段程序的输出结果是

class CParent {

public:

CParent() { Print(); };

virtual ~CParent() {};

virtual void Print() = 0;

}

class CSon : public CParent {

public:

CSon() { Print(); };

virtual ~CSon() {};

Print() { std::cout<<”2”; };

}

int main()

{

CParent\* pParent = new Son();

delete pParent;

}

A. 2,2

B. 2

C. 输出结果不确定

D. 以上都不对

18.（单选）以下代码运行结果是

#include <iostream>

using namespace std;

#define ALPHA(x,y) x##y

#define BETA beta

#define GAMMA gamma

int main()

{

int ALPHA(BETA, GAMMA) = 1;

cout<<BETAGAMMA;

return 0;

}

A. 1

B. BETA GAMMA

C. BETAGAMMA

D. 编译错误

19. \*\*（多选）假如有std::vector<int>类型的容器coll，需要对其元素从大到小排列，下列写法正确的有

A. std::sort(coll.begin(), coll.end(), std::greater<int>());

B. std::sort(coll.begin(), coll.end(), [](int a, int b) { return a >b; });

C. struct Cmp {

bool operator() (int a, int b) const {

return a > b;

}

};

std::sort(coll.begin(), coll.end(), Cmp);

D. bool Cmp(int a, int b)

{

return a > b;

}

std::sort(coll.begin(), coll.end(), Cmp);

20. \*\*（单选）以下代码输出的是

#include <iostream>

using namespace std;

class Foo {

public:

Foo() = default;

Foo(const Foo& foo) {

cout<<"复制构造函数:Foo(const Foo& foo)"<<endl;

}

Foo(Foo&& foo) {

cout<<"移动构造函数:Foo(Foo&& foo)"<<endl;

}

Foo& operator=(const Foo &foo) {

cout<<"赋值运算符:Foo& operator=(const Foo &foo)"<<endl;

return \*this;

}

Foo& operator=(Foo &&foo) {

cout<<"移动赋值运算符Foo& operator=(Foo &&foo)"<<endl;

return \*this;

}

};

Foo GetFoo()

{

Foo foo;

return foo;

}

void Pass(Foo foo)

{

;

}

Foo PassAndReturn(Foo foo)

{

return foo;

}

int main()

{

Foo foo1, foo2;

cout<<"--------------------------------"<<endl<<"1:\n";

foo1 = GetFoo();

cout<<"--------------------------------"<<endl<<"2:\n";

foo2 = foo1;

cout<<"--------------------------------"<<endl<<"3:\n";

foo2 = PassAndReturn(foo1);

cout<<"--------------------------------"<<endl<<"4:\n";

Pass(foo1);

cout<<"--------------------------------"<<endl<<"5:\n";

Foo foo3 = PassAndReturn(foo1);

cout<<"--------------------------------"<<endl;

return 0;

}

21. 以下关于c++的 bss和data 数据段说法正确的是

A 未初始化的变量放在 bss,

B 初始化非零的 const 变量放在 data

C 初始化非零的 const 变量放在 bss

D 初始化为0 的全局变量放在 bss

22. 下面代码运行结果是（）

#include <iostream>

#include <tuple>

int main() {

int a = 1, b =5;

int &r = a;

auto t = std::tie(r,b);

std::get<0>(t) = 3;

std::get<1>(t) += 2;

cout<<a<<" "<<b<<endl;

return 0;

}

A. 1 5

B. 3 7

C. 1 7

D. 3 5

23. 以下均为涉及动态内存管理的代码，分配的内存都只在局部使用，return之前要保证释放。其中符合公司C++通用编程规范的规则和建议的是（）

A.

Image\* imageA = new Image;

Image\* imageB = new Image;

// …

if (ProcFail()) {

delete imageA;

delete imageB;

return;

}

// …

delete imageA;

delete imageB;

return;

B.

auto imageA = std::make\_unique();

auto imageB = std::make\_unique();

if (ProcFail()) {

return;

}

return;

C.

auto imageA = std::make\_shared();

auto imageB = std::make\_shared();

// …

if (ProcFail()) {

return;

}

//…

return;

D.

std::auto\_ptr imageA(new image);

std::auto\_ptr imageB(new image);

// …

if (ProcFail()) {

return;

}

// …

return;

24. 下面使用模板参数推导C++17代码中哪一项是错误的

A auto pr = std::make\_pair(3, 5);

B std::pair pr(3, 5);

C std::array a{3, 5};

D std::vector v{3.0, 5};

25.（单选）每个变量的所有取值，能覆盖一次的是（）

A EC 单一组合

B BC 基本组合

C N-wise

D pair-wise

26. （单选）下列示例代码中对于枚举的使用方法，符合公司华为C++语言编程规范的是

A.

#define BOARD\_COUNT 64;

OspBoard neBoards[BOARD\_COUNT]

B.

enum WeekDay {

　　SUNDAY\_ZERO = 0,

　　MONDAY\_ONE = 1,

　　TUESDAY\_TWO = 2,

　　WEDNESDAY\_THREE = 3,

　　THURSDAY\_FOUR = 4,

　　FRIDAY\_FIVE = 5,

　　SATURDAY\_SIX = 6

};

C.

enum ScoreLevel {

　　BELOW\_AVERAGE = 50,

　　PASSED = 60,

　　GOOD = 80,

　　GROUP\_THRESHOLD = 80,

　　EXCELLENT = 90

};

D.

enum class ThreadState {

　　PENDING,

　　RUNNING,

　　WAITING,

　　SUSPENDED,

　　INVALID

}

enum class MsgState {

　　SENDING,

　　WAITING,

　　FINISHED,

　　INVALID

}

27. \*\*（单选）按照华为通用编程规范要求，应填入横线处的代码为

class Base {

public:

virtual ~Base() {}

};

class Derived : public Base {

public:

virtual ~Derived() {}

};

int Func(Base\* object)

{

Derived\* derived = \_\_\_\_\_\_;

if (derived == nullptr) {

reurn ERROR\_WRONG\_TYPE;

}

}

A. (Derived\*)object;

B. dynamic\_cast<Derived\*>(object);

C. static\_cast<Derived\*>(object);

D. reinterpret\_cast<Derived\*>(object);

28. （单选）阅读下面代码，输入“abcdef”，请问输出结果

char str[10];

cin.get(str, 5, 'd');

cout << str;

A. abc

B. abcd

C. abcde

D. abcdef

29.【存疑】foo.h中定义了模板类Foo，那么对于template<>class Foo(std::string)这个模板显示特殊化定义的位置，符合规范的是（）

template <typename T>

class Foo{

}

A、 必须放在foo.h中

B、 可以放在foo.h或其他包含了foo.h的头文件中

C、 必须放在.cpp中

D、 必须放在同时包含了foo.h和<string>的某个cpp文件中

30. \*\*\*以下输出？

cout << "Alpha\0and\0Gamma";

1. Alpha
2. Alpha\0and\0Gamma
3. Gamma
4. 无输出

31. \*\*对std::vector<int>类型的变量，以下符合华为语言编程规范的原则，要求和建议的是（）

A. for (size\_t i = 0; i < v.size(); ++i) {

B. for (int i = 0; i < v.size(); ++i) {

C. for (auto i = 0; i < v.size(); ++i) {

D. for (auto i{0}; i < v.size(); ++i) {

32.(单选)根据华为C++语言编程规范，针对cpp文件中不需要导出的变量，常量或者函数，下列哪种说法是正确的？

A. 只能使用 static 修饰

B. 只能使用匿名 namespace 封装

C. 可以使用匿名namespace封装或static修饰，推荐使用匿名namespace

D. 可以使用匿名namespace封装或static修饰，推荐使用static

33.左值引用捕获异常

34. （多选）代码的注释处可以插入的语句有（）

class Alpha{

public:

char factor;

Alpha(char value):factor(value){}

};

int main()

{

Alpha a1=Alpha(‘a’);

Alpha a2=Alpha(‘b’);

Alpha a3=Alpha(‘c’);

Alpha \*t[]={&a,&b,&c};

for(Alpha \*\*i=t;t+2!=i;i++){

————————

}

return 0;

}

A. std::cout <<(\*i)->factor;

B. std::cout <<(\*\*i).actor;

C. std::cout <<i.actor;

D. std::cout <<\*i->factor;

35.匿名函数捕获题

36. （单选）下面auto推导结果错误的有哪些（）

A.

auto a = 1; // 推导出变量 a 的类型是 int。

B.

int a = 1；

auto b = a; // 推导出变量 b 的类型是 int。

C.

int a =1;

const auto &c = a; //推导出变量 c 的类型是 const int& 。

D.

const int a =1;

auto d = a; //推导出变量 d 的类型是const int。

37.（单选）下面代码中，符合华为C++语言编程规范的是（）

A.

#define BOARD\_COUNT 64;

OspBoard neBoards[BOARD\_COUNT]

B.

enum WeekDay {

　　SUNDAY\_ZERO = 0,

　　MONDAY\_ONE = 1,

　　TUESDAY\_TWO = 2,

　　WEDNESDAY\_THREE = 3,

　　THURSDAY\_FOUR = 4,

　　FRIDAY\_FIVE = 5,

　　SATURDAY\_SIX = 6

};

C.

enum ScoreLevel {

　　BELOW\_AVERAGE = 50,

　　PASSED = 60,

　　GOOD = 80,

　　GROUP\_THRESHOLD = 80,

　　EXCELLENT = 90

};

D.

enum class ThreadState {

　　PENDING,

　　RUNNING,

　　WAITING,

　　SUSPENDED,

　　INVALID

}

enum class MsgState {

　　SENDING,

　　WAITING,

　　FINISHED,

　　INVALID

}

38. （单选）\*\*调用foo函数的返回值为（）

char foo(void)

{

　　unsigned int a = 6;

　　int b = -20;

　　char c;

　　(a + b > 6) ? (c = 1) : (c = 0);

　　return c;

}

A. 运行出错 B. 编译错误 C. 1 D. 0

39. （单选）以下代码，宏打印结果为：

#define ALPHA(x) if (x) {cout << #x;} else {cout << '?';}

int main() {

int i = 1;

ALPHA(i < 3)

return 0;

}

1. i<3
2. #x
3. 0
4. ?

40.（多选）测试边界值分析，利用哪些边界数值作为测试数据？（）

A.最大值最小值

B.最大值+1

C.最小值-1

D.默认值

41. （单选）哪个是定义的指向10个指针元素的指针（）

A、int\* p[10];

B、int (\*p)[10];

C、int\* (\*p)[10];

D、int (\*p[10])(int\*);

42.（单选）问以下代码输出是什么：（）

class Base {

public:

Base(){}

~Base(){}

void\* operator new(size\_t n) {

cout<<” operator new 1” <<endl;

return malloc(n);

}

void\* operator new[](size\_t n) {

cout<<” operator new 2” <<endl;

return malloc(n);

}

};

void\* operator new(size\_t n) {

cout<<” operator new 3” <<endl;

return malloc(n);

}

void\* operator new(size\_t n,char\* c,int line) {

cout<<” operator new 4” <<endl;

return malloc(n);

}

int main()

{

Base \*b = new Base();

return 0;

}

1. operator new 1
2. operator new 2
3. operator new 3
4. operator new 4

43.（单选 题目考点完全相同）lambda表达式的输出（）

int main()

{

int a = 12;

int b = 13;

auto f1 = [=](){cout<<a+b<<endl;};

auto f2 = [&](){cout<<a+b<<endl;};

auto f3 = [a,&b](){cout<<a+b<<endl;};

auto f4 = [&a,b](){cout<<a+b<<endl;};

b++;

f1();

f2();

f3();

f4();

return 0;

}

44（单选）运行结果（）

string getString();

void main(){

string str = "gamama";

cout << getString(str) << endl;

}

string getString(string &str){

return str;

}

A. gamama

B. g

C. 编译错误

D. 运行错误

45.（多选）怎么让这段代码输出012？

int select = 0;

switch (select)

{

case 0: cout<<0;

break;

case 1: cout<<1;

break;

default: cout<<2;

break;

case 2: cout<<3;

break;

}

A. 删除第一个break

B. 删除第二个break

C. 删除default分支

D. 删除第三个break

46. （单选）\*\*\* 正确的函数返回值方式：（）

A.

S Func{

S result;

return result;

}

B.

S\* Func{

S result;

return &result;

}

C.

S& Func{

S result;

return std::move(result);

}

D.

S&& Func{

S result;

return std::forward<S>(std::move(result));

}

47.异常捕获 ，选const &

try {throw BarException();} catch (const FooException& e) { ... }

48.（单选）命名空间求输出

#include <iostream>

using namespace std;

int a = 1;

namespace Outer {

int a = 2;

}

namespace Outer {

namespace Inner {

int a = 3;

}

}

using namespace Outer;

int main()

{

int a = 4;

cout << a;

cout << ::a;

cout << Outer::a;

cout << Outer::Inner::a;

return 0;

}

49. （单选）请问以下代码的输出为

int main()}{

map<string ,int> mp;

mp.insert({"abc",3});

mp.insert({"def",1});

mp.insert({"abcdef",2});

for(auto iter = mp.begin();iter !=mp.end(); iter++){

cout<< iter->second<<endl;

}

return 0;

}

A 1 2 3

B 3 2 1

C 2 3 1

D 2 1 3

50. （单选）函数Print的功能是对自定义类型MyClass中的值进行调试打印，不会对MyClass对象进行任何修改，则下列函数声明中符合华为C++语言编程规范的是

A. void Print(std::unique\_ptr<MyClass> obj);

B. void Print(const std::unique\_ptr<MyClass>& obj);

C. void Print(std::shared\_ptr<MyClass> obj);

D. void Print(const MyClass& obj);

51.（单选）以下哪条命令可以正确设置条件断点（test\_case.c为文件名，TestCase为函数名，isInit为Testcase中的局部变量）（）

A. break test\_case.c:100 while (isInit != 0)

B. break test\_case.c:100 if (isInit != 0)

C. break TestCase:100 while (isInit != 0)

D. break test\_case.c:100 (isInit != 0)

52.（单选）软件有多个测试输入，对每个输入组合设计用例，每种输入的每个值都能出现在组合中，应当选用哪种方法设计用例？

A、EC(单一选择)

B、BC(基本选择)

C、N-wise

D、pair-wise

53. （单选）关于断言，下列符合规范的是()

A、if(sizeo(int)!=4){

abort();

}

B、assert(sizeof(int))==4);

C、ASSERT (sizeof(int)==4)

D、static\_assert(sizeof(int)==4)

54. （单选）如果因为某些原因需要使用sprintf\_s函数，符合关于格式化输入/输出函数的要求是（）x,y均为int类型，buf为元素类型为char的数组

A、sprintf\_s(buf,sizeof(buf),”%d”,x);

B、sprintf\_s(buf,sizeof(buf),”%d”,x,y);

C、sprintf\_s(buf,sizeof(buf),”%d”);

D、sprintf\_s(buf,sizeof(buf),” ”,x);

55. （单选）以下两段从代码属于两个不同的程序，在每个程序中，当proc被多线程并发调用了10000次后，对于相应的全局变量的值，说法正确的是（）

std::atomic<std::uint32\_t> g\_a(0);

void proc()

{

++g\_a;

}

volatile std::uint32\_t g\_v(0);

void proc()

{

++g\_v;

}

A、 g\_a的值为10000

g\_v的值不确定

B、 g\_a的值不确定

g\_v的值为10000

C、 g\_a的值为10000

g\_v的值为10000

D、 g\_a的值不确定

g\_v的值不确定

56. （新题只记住了选项） gtest单元测试，一段代码，让填入合适的代码

A. TEST\_P(addList, result);

B. TEST\_P(addListCase, result);

C. TEST\_F(addListCase, result);

D. TEST\_F(addList, result);

57. 【多选】不保证对齐，下列指针转换符合C++编程规范继承自C语言规范条款的有（）【存疑】

A.

char \*pc;

int \*pi = reinterpret\_cast<int \*>(pc);

B.

int \*pi;

char \*pc = reinterpret\_cast<char \*>(pi);

C.

char \*pc;

void \*pv = pc;

int \*pi = static\_cast<int \*>(pv);

D.

int \*pi;

void \*pv = pi;

float pf = static\_cast<float \*>(pv);

58. 【单选】【tie】下面代码运行结果是（）

int a = 1, b =5;

int &r = a;

auto t = std::tie(r,b);

std::get<0>(t) = 3;

std::get<1>(t) += 2;

cout<<a<<" "<<b<<endl;

A. 1 5

B. 3 5

C. 1 7

D. 3 7

59.（新题只记得选项）一段代码，让选择合适的代码插入其中，使得不会发生数据竞争

题目给出了async的c++标准定义

A. std::async(std::launch::deferred……

B. std::async(std::launch::async……

C. std::async(std::launch::async | std::launch::deferred)

D. std::async();

60.（新题多选）类内的public说法错误的是

A. 只作用于成员变量

B. 只作用于成员函数

C. 用于访问控制

D. 一个类中只能出现一个public

61.（新题单选）运行调用栈后的输出是下面三行代码，问真正运行代码的函数调用顺序是什么

/\*三行输出

1 ……func()……(还有行号)

2 ……test()……(还有行号)

3 …………main()…….(有文件名和行号)

\*/

A.func() - test() – main()

B.main() – test() – func()

C.test() – func()

D.func() – test()

62. 我们在设计一个雇员关系的类图，其中雇员(Employee)类中有的实例是其它实例的主管，比如有雇员类的实例a、b、c，其中c是a、b的主管，应该使用下面哪种关系表示

A. 自身关联（Reflexive Association）

B. 聚合（Aggregation）

C. 泛化（Generalization）

D. 定向关联 (Directed Association)

63. 在软件需求工程中，需求管理贯穿整个过程。需求管理最基本的任务是明确需求，使项目团队和用户达成共识，建立

A. 需求跟踪说明

B. 需求变更管理文档

C. 需求分析计划

D. 需求基线

64．需求描述要清晰、需求可验收。小明看到以下关于设备数据采集功能相关的需求，请你帮忙审核下，需求满足清晰、 可验收标准的是

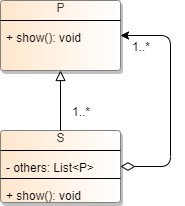
A．设备支持温度、功耗等信息实时统计

B．数据支持FTP协议传输

C．支持新老特性兼容

D．数据采集性能提升10%左右

65．请选择出下面类图中存在的问题（ ）



A. S类与P类的继承关系中不能指定多重性（1..\*）

B. S类与P类的聚合关系中不能指定多重性（1..\*）

C. S类与P类已经有继承关系，不能再存在聚合关系

D. S类不能与P类有重复的成员函数show

66. 针对重构方法，下列说法错误的有（）

A.提取接口（ExtractInterface）和提取超类（ExtractSuperclass）均可提取共通代码

B.隐藏委托关系（HideDelegate）会使代码层次更清晰，因此委托类的功能越多越好

C.当发现某个子类并未带来该有的派生价值时，可以考虑使用移除子类（RemoveSubclass）手法进行重构

D.将值域上移到父类（即字段上移Pull UpField）会有效减少子类的成员变量，只需要般移值域，不需要般移对该值域的操作方法

67. 静态分析检查是静态测试的一种方法，下面哪些是静态分析能够发现的缺陷？()

A. 引用未定义的变量

B. 从未使用的变量

C. 不可达的代码

D. 规格说明书的错误

E. 代码语法错误

68. 需求跟踪是需求管理活动之一，保证每项需求都能被关联跟踪，包括以下哪些关联

A、与设计关联

B、与代码关联

C、与测试关联

D、与需求状态关联