

Вариант №109

Группа: 1304

Группа: 1304      ФИО: Чернякова Валерия Алексеевна

1-(0.5): Выберите верные утверждения об инкапсуляции:

- а) Позволяет классам наследникам работать с полями базового класса
- б) Позволяет классам наследникам перенимать свойства родительского класса
- в) Это сокрытие данных
- г) Все поля и методы классы должны быть private для соблюдения принципа

2-(0.5): При каком модификаторе доступа полей класса, прямой доступ к ним имеют любые другие классы:

- а) public  
б) такого модификатора нет  
в) private  
г) protected

3-(0.5): В каких случаях в функцию можно передать lvalue-значение:

- ☐ a) void func(int&& value)  
☐ b) void func(const int& value)  
☐ B) void func(int\* value)  
☒ d) void func(const int&& value)

4-(0.5): Что такое перегрузка оператора?

4-(0.5): Что такое перегрузка оператора? *Перегрузка оператора — переопределение его свойств, параметров, что и, целое, ввиду?*

5-(0.5): Выберите верные утверждения о статических методах класса:

5-(0.5): Выберите верные утверждения о статистических методах класса:

- а) Статические методы не могут взаимодействовать с не статическими полями класса  
 б) Статические методы могут быть вызваны через идентификатор имени класса  
 в) Статические методы не могут быть вызваны через объект класса  
 г) Статические методы могут быть вызваны через объект класса

6-(0.5): Какой из вариантов является сигнатурой деструктора класса `41sum`?

- a) ~41sum(void\*)  
b) delete(const 41sum&)  
B) 41sum()  
r) delete 41sum()

7-(0.5): Какой из вариантов является корректной сигнатурой перегрузки оператора вывода в поток для класса `Korvpiklaani`?

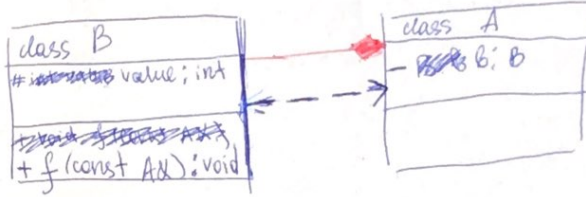
- a) ostream& operator<<(const Korpiklaani&  
b) istream& operator<<(istream&,const Korpiklaani&  
**B) ostream& operator<<(ostream&,const Korpiklaani&**  
r) ostream& operator>>(ostream,const Korpiklaani&)

8-(1): Что такое конструктор перемещения?

Конструктор перемещения - перемещение объектов при их указании.  
 сигнатура `Someclass (Someclass& obj)`  
 при реализации необходимо переопределение оператора прибавления.  
 работает на `std::move` и `swap` семантике

9-(1): Постройте UML-диаграмму классов для следующего кода

```
class B{
protected:
    int value;
public:
    void f(const A&);
};
class A{
    B b;
};
```



10-(1): Что такое множественное наследование? Когда один класс является производным / дочерним сразу от нескольких классов.

11-(1): Что такое полиморфный класс, какие виды полиморфных классов есть? Класс, который имеет хотя бы один виртуальный метод.

12-(1): В каких случаях static\_cast вернет nullptr?

- a) Если передать объект не полиморфного класса
- ☒ б) Если приводить nullptr к указателю на какой-то тип
- ☒ в) Когда нет нужной функции приведения к типу
- ☒ г) static\_cast не возвращает nullptr

13-(1): Какие ошибки в следующем коде?

```
class A{
    int state;
public:
    A(int state = 0, bool flag): state(state) {} // вызов конструктора, в котором объявлен не сущ-во поле класса
};
class B{
public:
    void proceed(A& obj){ // вызов метода, в котором объявлен не сущ-во поле другого класса, хотя
        ++obj.state; // мы не имеем к нему доступ.
    }
};
```

14-(0.5): Как исправить ошибки в коде из задания №13 вашего варианта?

15-(0.01): Какая музыкальная группа лучше?

- a) Meshuggah
- ☒ б) Grima
- ☒ в) Korpiklaani
- г) And So I Watch You From Afar

```
4) class A {
    friend class B;
    int state;
public:
    A(int state = 0): state(state) {}
};
class B {
public:
    void proceed(A& obj) {}
};
void proceed(A& obj) {
    ++obj.state;
};
```