Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.3**

Дисциплина: «Информатика»

Тема: Объектно-ориентированное программирование. Перегрузка операций.

Вариант 13

Выполнил:

Студент группы РИС-20-1б

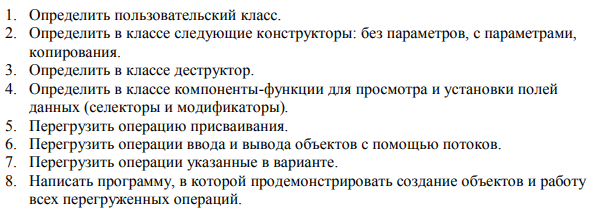
Кузнецов Михаил Сергеевиич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Постановка задачи**

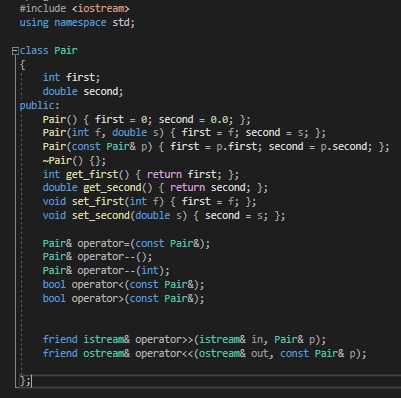


**Анализ задачи**

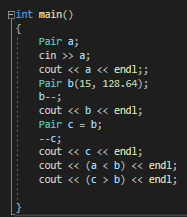
1. Для решения задачи необходимо…
   1. Организовать класс Pair, первое целое число и второе вещественное.



* 1. Организовать необходимые методы для ввода данных в поля first и second: гетторы, сетторы, конструкторы, деструктор, перегрузки операторов.



* 1. В функции main произвести тестирование.



1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения результата первого числа;



* 1. Тип данных double для хранения второго числа;



**Код**

#include <iostream>

#include "Pair.h"

using namespace std;

int main()

{

Pair a;

cin >> a;

cout << a << endl;;

Pair b(15, 128.64);

b--;

cout << b << endl;

Pair c = b;

--c;

cout << c << endl;

cout << (a < b) << endl;

cout << (c > b) << endl;

}

#include "Pair.h"

#include <iostream>

using namespace std;

Pair& Pair::operator=(const Pair& p)

{

if (&p == this) return \*this;

first = p.first;

second = p.second;

return \*this;

}

Pair& Pair::operator--()

{

first--;

return \*this;

}

Pair& Pair::operator--(int)

{

second--;

return \*this;

}

bool Pair::operator<(const Pair& p)

{

return (first + second < p.first + p.second);

}

bool Pair::operator>(const Pair& p)

{

return (first + second > p.first + p.second);

}

istream& operator>>(istream& in, Pair& p)

{

cout << "Enter first: ";

in >> p.first;

cout << endl;

cout << "Enter second: ";

in >> p.second;

cout << endl;

return in;

}

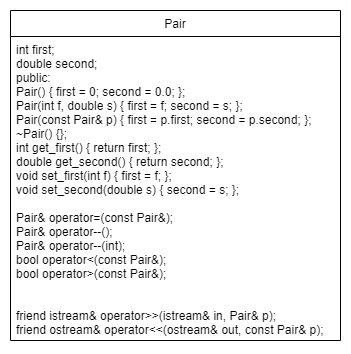
ostream& operator<<(ostream& out, const Pair& p)

{

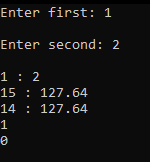
return (out << p.first << " : " << p.second);

}

**UML диаграмма**

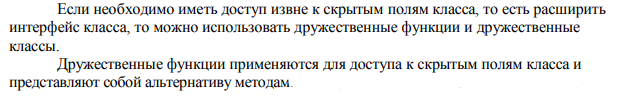


**Скриншоты тестов**

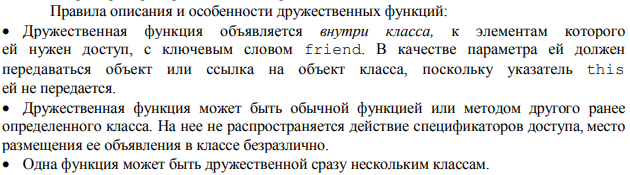
****

**Контрольные вопросы**



1. 

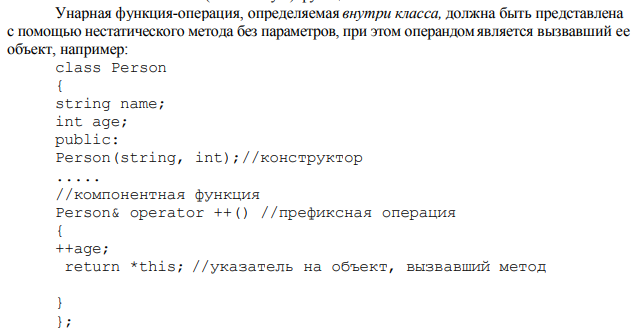


2. 

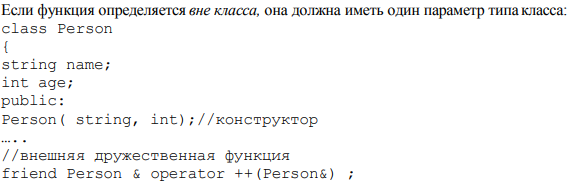


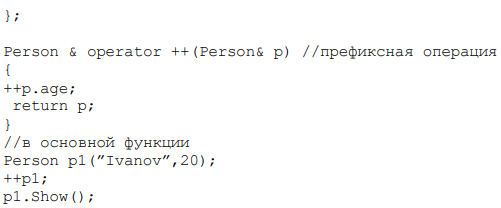
3. 



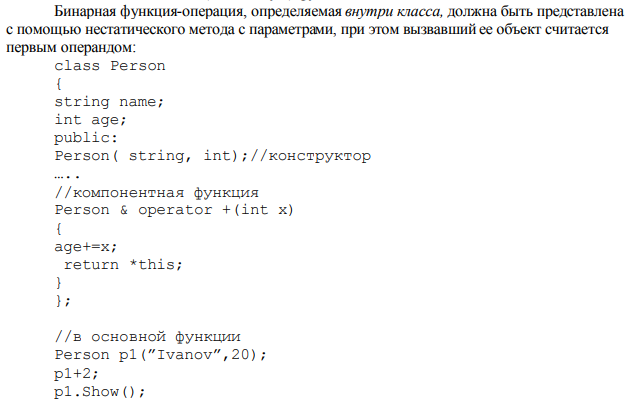
4.



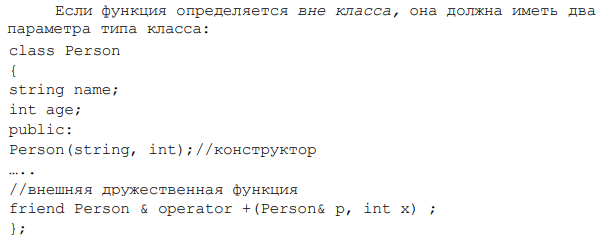
5.





6.

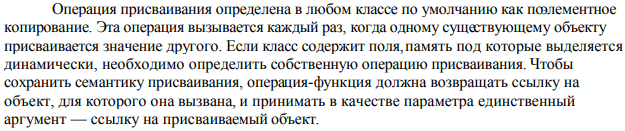


7. 



8. 

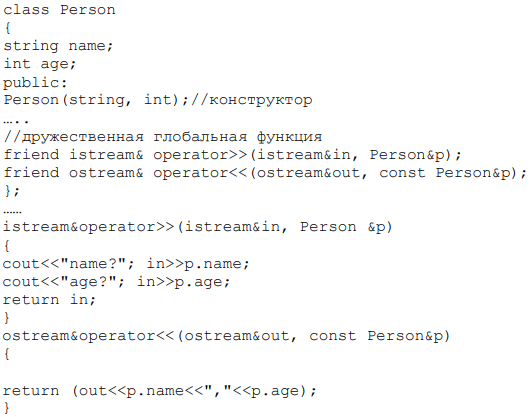


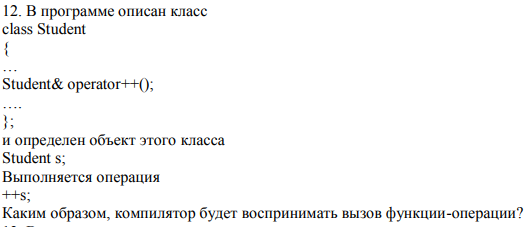
9. 



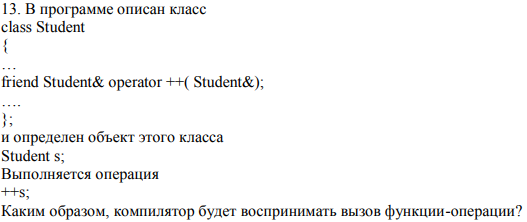
10. Ссылку на объект, для которого операция была вызвана.



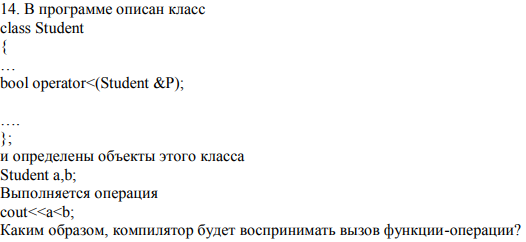




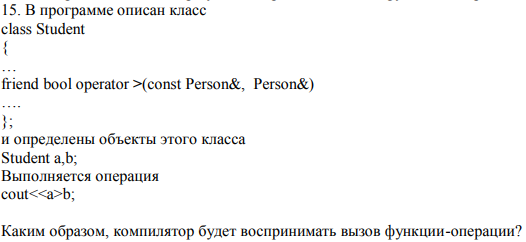
12. Будет осуществлён вызов функции перегрузки оператора префиксного инкремента.



13. Будет осуществлён вызов функции перегрузки оператора постфиксного инкремента.



14. Будет осуществлён вызов функции перегрузки оператора сравнения (<), затем через cout будет выведен результат сравнения (1 или 0).



15. Будет осуществлён вызов функции перегрузки оператора сравнения (>), затем через cout будет выведен результат сравнения (1 или 0).