Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа 3.**

**" Классы и объекты. Перегрузка операторов"**

Выполнила студентка гр. РИС-24-3б

Черепанова Диана Романовна

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

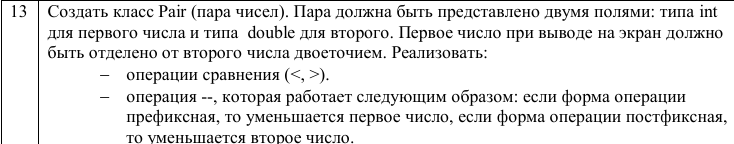
(оценка) (подпись)

(дата)

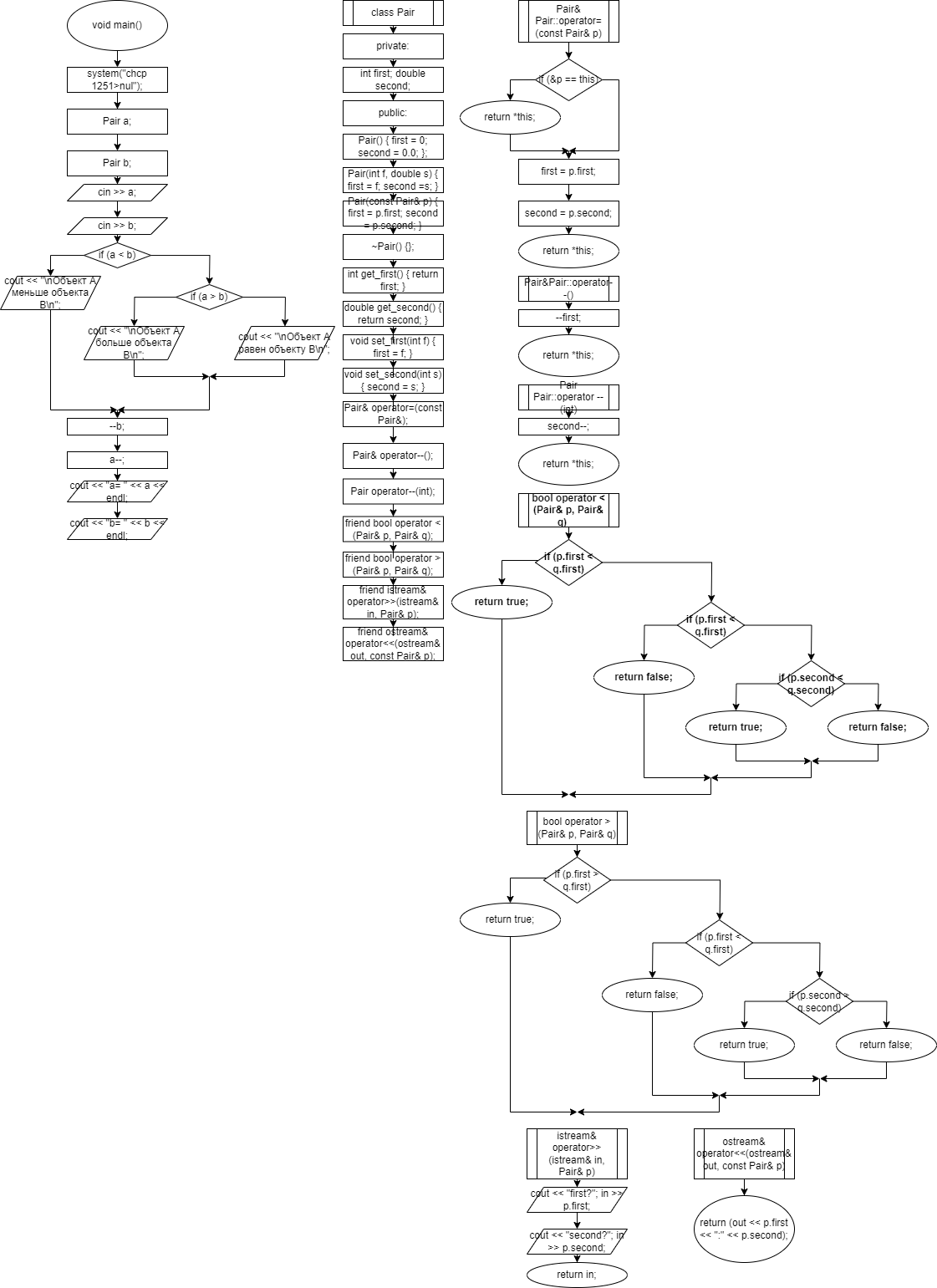
г. Пермь, 2025

**Постановка задачи**

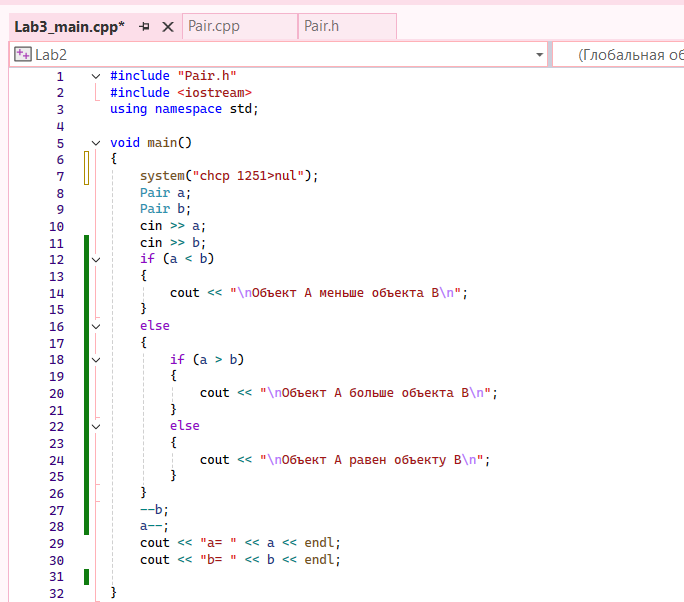
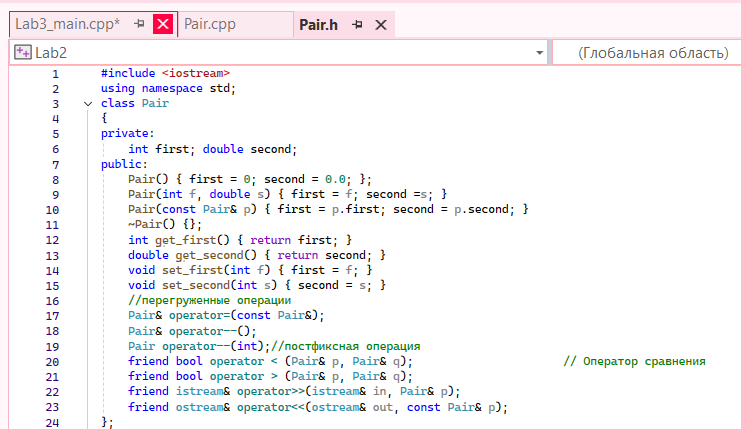
1. Определить пользовательский класс
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).
5. Перегрузить операцию присваивания.
6. Перегрузить операцию ввода и вывода объектов с помощью потоков.
7. Перегрузить операции, указанные в варианте.
8. Написать демонстрационную программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.



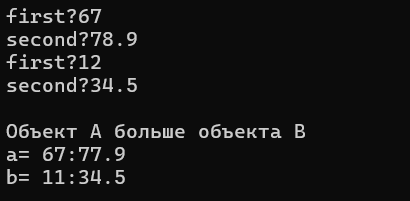
Блок-схема



Код

Результат программы



Ответы на вопросы

1. Для чего используются дружественные функции и классы?

Дружественные функции применяются для доступа к скрытым полям класса и представляют собой альтернативу методам. Дружественный класс применяется для доступа методов одного класса к скрытым полям другого.

1. Сформулировать правила описания и особенности дружественных функций.

* Объявление дружественной функции – внутри нужного класса, ключевым словом friend. Передаваемый параметр - объект или ссылка на объект.
* Дружественная функция – обычная функция или метод другого класса. Не важно, где функция описана в классе + спецификаторы доступа не распространяются.
* Одна и та же дружественная функция может быть в нескольких классах.

*Пример*

class student;//предварительное описание класса

class teacher{

…

void teach(student &S);

….

};

class student{

...

friend void teacher::teach(student&);//дружественная функция,

//имеет доступ к элементам класса student

…

};

1. Каким образом можно перегрузить унарные операции?

Унарную операцию можно перегрузить как:

* Компонентная функция класса
* Внешняя (глобальная) функция

Синтаксис перегрузки:

тип operator знак\_оператора(void);

*Пример*

class Person{

string name;

int age;

public:

Person(string, int);//конструктор

//компонентная функция //префиксная операция

Person& operator ++() {

++age;

return \*this; } //указатель на объект, вызвавший метод

};

//в основной функции

Person p1(”Ivanov”,20);

++p1;

p1.Show();

* Вне класса (как внешнюю функцию). Перегружается как функция с параметром типа класса.

Синтаксис перегрузки:

тип operator знак\_оператора(class A); //передача объекта класса

*Пример*

class Person {

string name;

int age;

public:

Person(string, int);//конструктор

friend Person & operator ++(Person&) ; //внешняя дружественная функция

};

Person & operator ++(Person& p) { //префиксная операция

++p.age;

return p; }

//в основной функции

Person p1(”Ivanov”,20);

++p1;

p1.Show();

1. Сколько операндов должна иметь унарная функция-операция, определяемая внутри класса?

Унарная функция-операция, определяемая внутри класса, не имеет операндов.

1. Сколько операндов должна иметь унарная функция-операция, определяемая вне класса?

Унарная функция-операция, определяемая вне класса, имеет один операнд – ссылка на вызвавший её объект.

1. Сколько операндов должна иметь бинарная функция-операция, определяемая внутри класса?

Бинарная функция-операция, определяемая внутри класса, имеет один операнд – переменная, хранящая значение.

1. Сколько операндов должна иметь бинарная функция-операция, определяемая вне класса?

Бинарная функция-операция, определяемая вне класса, имеет два операнда. Первый – вызвавший её объект, второй – переменная, хранящая значение.

1. Чем отличается перегрузка префиксных и постфиксных унарных операций?

Перегрузка префиксной унарной операции не имеет передаваемых параметров, перегрузка постфиксной унарной операции имеет один передаваемый параметр – показатель того, что это постфиксная форма.

1. Каким образом можно перегрузить операцию присваивания?

Перегрузка операции присваивания описывается как метод класса. В качестве принимаемого аргумента используется ссылка на присваиваемый объект, в качестве возвращаемого значения – ссылку на объект, для которого она вызвана.

class Person {

string name;

int age;

public:

Person(string, int);//конструктор

Person& operator =(const Person&); //компонентная функция

};

Person& Person::operator = (const& Person p) {

//проверка на самоприсваивание

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

name = p.name;

age = p.age;

return \*this; }

//в основной функции

Person p1(”Ivanov”,20);

Person p2;

p2=p1;

p1.Show();

p2.Show();

1. Что должна возвращать операция присваивания?

Операция присваивания возвращает ссылку на объект, для которого она вызвана.

1. Каким образом можно перегрузить операции ввода-вывода?

Перегрузка операций ввода-вывода описываются как дружественные функции, т. к. левый операнд – потоки ввода-вывода.

1. В программе описан класс

class Student

{

……

Student& operator ++();

……

};

И определён объект этого класса

Student s;

Выполняется операция

++s;

Каким образом, компилятор будет воспринимать вызов функции-операции?

Компилятор воспринимает данный вызов как перегруженную функцию-операцию префиксного инкремента, определенную внутри класса.

1. В программе описан класс

class Student

{

……

friend Student& operator ++(Student&);

……

};

И определён объект этого класса

Student s;

Выполняется операция

++s;

Каким образом, компилятор будет воспринимать вызов функции-операции?

Компилятор воспринимает данный вызов как дружественную перегруженную функцию-операцию префиксного инкремента.

1. В программе описан класс

class Student

{

……

bool operator < (Student& p);

……

};

И определены объекты этого класса

Student a, b;

Выполняется операция

cout << a < b;

Каким образом, компилятор будет воспринимать вызов функции-операции?

Компилятор воспринимает данный вызов как перегруженную функцию-операцию оператора сравнения, определенную внутри класса.

1. В программе описан класс

class Student

{

……

friend bool operator > (const Person&, Person&);

……

};

И определён объект этого класса

Student a, b;

Выполняется операция

cout << a > b;

Каким образом, компилятор будет воспринимать вызов функции-операции?

Компилятор воспринимает данный вызов как дружественную перегруженную функцию-операцию сравнения.

GitHub: https://github.com/d1frd