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: 09.03.01 - «Информатика и вычислительная техника»

**Отчёт по лабораторной работе “Классы №1”**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Волков Роман Алексеевич

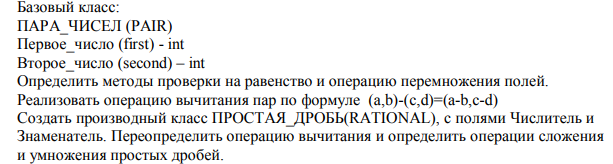
Проверил:

Яруллин Денис Владимирович

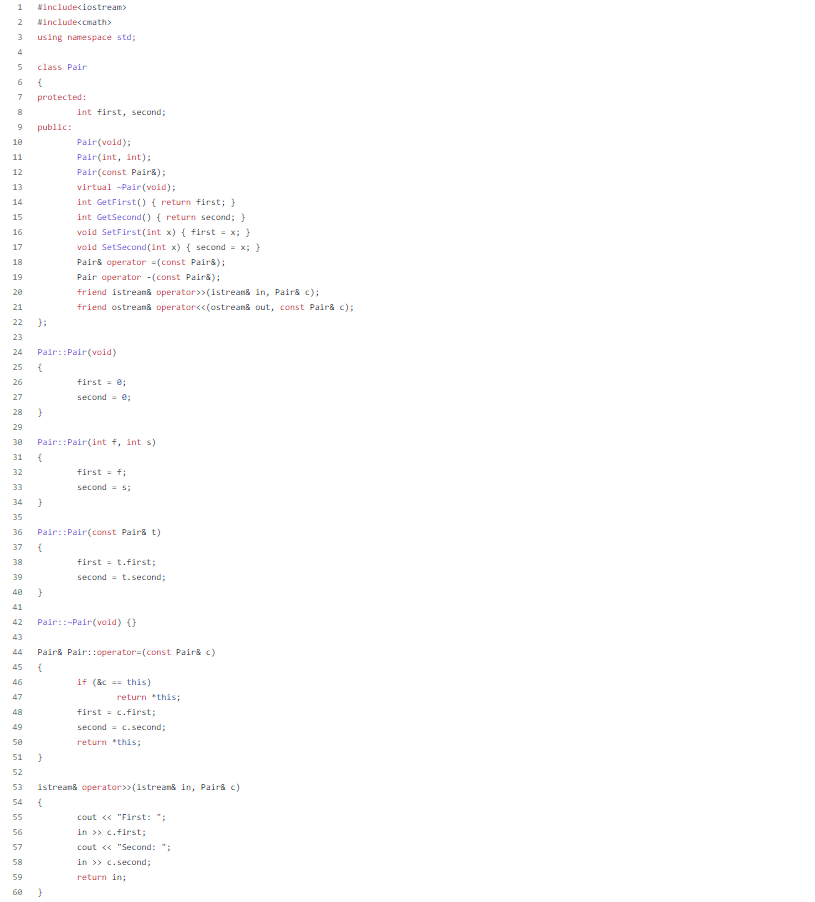
(подпись) (оценка)

г. Пермь-2022

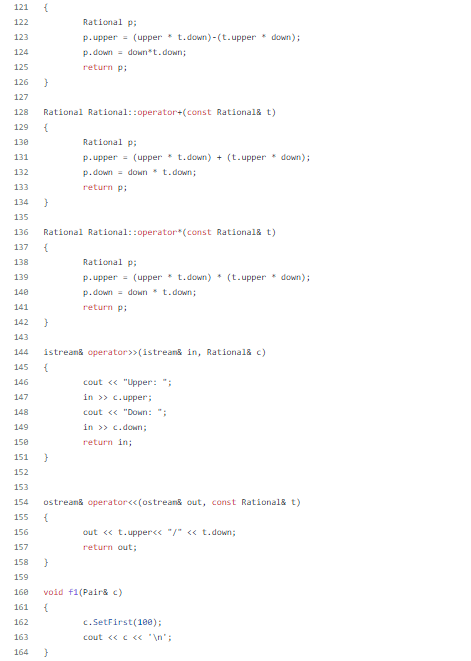
**Постановка задачи:**

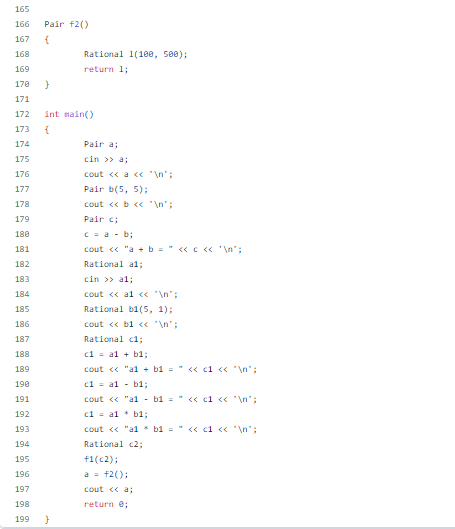


**Код программы:**









**Контрольные вопросы:**

1)

Для чего используется механизм наследования?

Наследование - это механизм получения нового класса на основе уже

существующего. Существующий класс может быть дополнен или

изменен для создания нового класса.

2)

Каким образом наследуются компоненты класса, описанные со

спецификатором public?

public – член класса может использоваться любой функцией, которая

является членом данного или производного класса, а также к public -

членам возможен доступ извне через имя объекта.

3)

Каким образом наследуются компоненты класса, описанные со

спецификатором private?

private – член класса может использоваться только функциями –

членами данного класса и функциями – “друзьями” своего класса. В

производном классе он недоступен.

4)

Каким образом наследуются компоненты класса, описанные со

спецификатором protected?

protected – то же, что и private, но дополнительно член класса с

данным атрибутом доступа может использоваться функциями-

членами и функциями – “друзьями” классов, производных от

данного.

5)

Каким образом описывается производный класс?

Производный класс наследует описание базового класса; затем он

может быть изменен добавлением новых членов, изменением

существующих функций-членов и изменением прав доступа. С

помощью наследования может быть создана иерархия классов,

которые совместно используют код и интерфейсы.

6)

Наследуются ли конструкторы?

Нет.

7)

Наследуются ли деструкторы?

Нет.

8)

В каком порядке конструируются объекты производных классов?

Объекты класса конструируются снизу вверх: сначала базовый,

потом компоненты- объекты (если они имеются), а потом сам

производный класс. Таким образом, объект производного класса

содержит в качестве подобъекта объект базового класса.

9)

В каком порядке уничтожаются объекты производных классов?

Уничтожаются объекты в обратном порядке: сначала производный, потом

его компоненты-объекты, а потом базовый объект.

10) Что представляют собой виртуальные функции и механизм позднего

связывания?

К механизму виртуальных функций обращаются в тех случаях,

когда в каждом производном классе требуется свой вариант

некоторой компонентной функции. Классы, включающие такие

функции, называются полиморфными и играют особую роль в ООП.

Виртуальные функции предоставляют механизм позднего

(отложенного) или динамического связывания. Любая нестатическая

функция базового класса может быть сделана виртуальной, для чего

используется ключевое слово virtual.

11) Могут ли быть виртуальными конструкторы? Деструкторы?

Конструкторы нет. Деструкторы да.

12) Наследуется ли спецификатор virtual?

Да.

13) Какое отношение устанавливает между классами открытое

наследование?

Private – не наследуется, protected – protected, public – public.

14) Какое отношение устанавливает между классами закрытое

наследование?

Private – не наследуется, protected – private, public – private.

15) В чем заключается принцип подстановки?

Открытое наследование устанавливает между классами отношение

«является»: класс-наследник является частью класса-родителя. Это

означает, что везде, где может быть использован объект базового

класса (при присваивании, при передаче параметров и возврате

результата), вместо него разрешается использовать объект

производного класса. Данное положение называется принципом

подстановки. Он работает и для ссылок и для указателей. Обратное

неверно. Например, всякий спортсмен (производный класс) является

человеком (базовый класс), но не всякий человек является

спортсменом.

Закрытое наследование – это наследование реализации, в этом

случае принцип подстановки не соблюдается.

16) Имеется иерархия

классов:

class Student

{

int age;

public:

string name;

...

};

class Employee : public Student

{

protected:

string post;

...

};

class Teacher : public Employee

{

protected: int stage;

...

};

Teacher x;

Какие компонентные данные будет иметь объект х?

Protected: int stage, string post; public: string name;

17) Для классов Student, Employee и Teacher написать конструкторы без

параметров.

Student() {age = 0; name = “”;}

Employee() {name = “”, post = “”;}

Teacher() {stage = 0; name = “”, post = “”;}

18) Для классов Student, Employee и Teacher написать конструкторы с

параметрами.

Student(int a, string n) {age = a; name = n;}

Employee(string n, string p) {name = n, post = p;}

Teacher(int s, string n, string p) {stage = s; name = n, post = p;}

19) Для классов Student, Employee и Teacher написать конструкторы

копирования.

Student(const Student& t) {age = t.age; name = t.name;}

Employee(const Employee& t) {name = t.name, post = t.post;}

Teacher(const Teacher& t) {stage = t.stage; name = t.name, post = p.post;}

20) Для классов Student, Employee и Teacher определить операцию

присваивания.

Student& operator =(const Student&c){

Age = c.age;

Name = c.name;

Return \*this;

}

Employee& operator =(const Employee&c){

post = c.post;

Name = c.name;

Return \*this;

}

Teacher& operator =(const Teacher&c){

Stage = c.stage;

Post = c.post;

Name = c.name;

Return \*this;

}