Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
**«Пермский национальный исследовательский политехнический университет»**

**(ПНИПУ)**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**Отчет о лабораторной работе №2**

Вариант №3

Дисциплина: основы алгоритмизации и программирования

Тема: классы в С++

Выполнил студент ИВТ-22-2б:

Якушев Михаил Витальевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Пермь 2023

**1 Постановка задачи**

1. Определить пользовательский класс.

2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

3. Определить в классе деструктор.

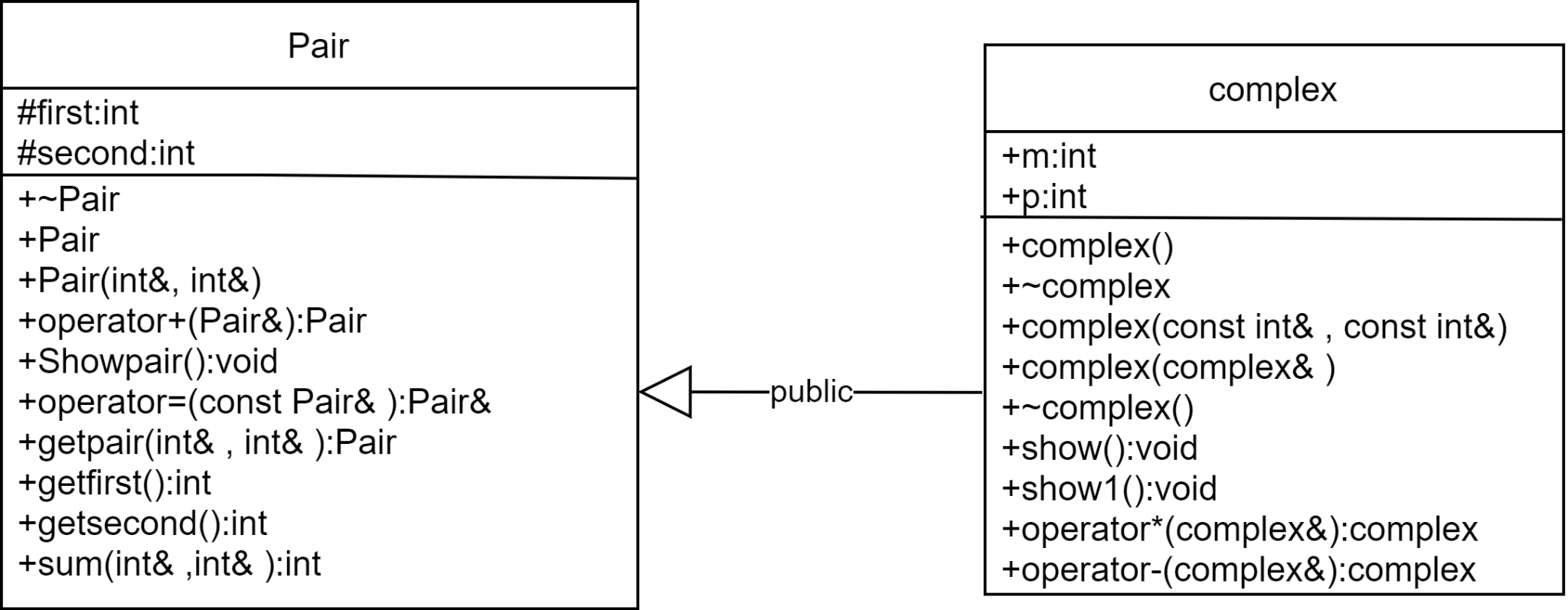
4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).

5. Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора-копирования, вызов конструктора с параметрами и конструктора без параметров.

**2 Анализ задачи**

Конструктор необходим для инициализации экземпляра класса. Конструктор – метод класса, имя которой совпадает с именем класса. По стандарту любой класс неявно содержит в себе конструктор без параметров и конструктор копирования. Конструктор копирования – создает новый объект на основе переданного ему в качестве параметра по ссылке. Все конструкторы использованы в данной программе.

**3 UML-диаграмма**

****

**4 Код программы**

#include <iostream>

#include<string>

using namespace std;

class A {

private:

string num;

int por;

float mant;

public:

A() {

cout << "Конструктор без параметров: " << endl;

this->mant = 0;

this->por = 0;

this->num = '-';

}

A(string c, int b, float a) {

cout << "Конструктор: " << endl;

this->por = b;

this->mant = a;

this->num = c;

}

A(A& tmp) {

cout << "Конструктор копирования: " << endl;

this->por = tmp.por;

this->mant = tmp.mant;

this->num = tmp.num;

}

void setpor(int por) {

this->por = por;

}

void setmant(float mant) {

this->mant = mant;

}

void setnum(string num) {

this->num = num;

}

int getpor() {

return por;

}

float getmant() {

return mant;

}

string getnum() {

return num;

}

void print() {

cout << num << endl;

cout << por << endl;

cout << mant << endl;

}

~A() {

cout << "Деструктор" << endl;

}

};

int main()

{

setlocale(LC\_ALL, "ru");

A num1;

num1.print();

A num2("252.15", 252, 0.15);

num2.print();

A num3 = num2;

num3.print();

num3.setnum("666.66");

num3.setpor(666);

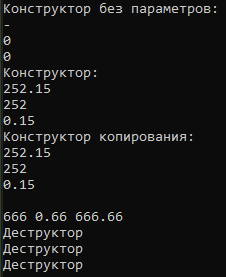
num3.setmant(0.66);

cout << endl << num3.getpor() << " " << num3.getmant() << " " << num3.getnum() << endl;

return 0;

}

**5 Результаты работы программы**

****

**6 Ответы на вопросы**

1. Для чего нужен конструктор?

Для инициализации параметров объекта класса.

2. Сколько типов конструкторов существует в С++?

3

3. Для чего используется деструктор? В каких случаях деструктор описывается явно?

Используется для удаления объекта. Описывается явно для освобождения динамической памяти.

4. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?

Без параметров обычно используется в качестве конструктора по умолчанию. Конструктор копирования используется для создания нового объекта, который создается путем копирования. Конструктор с параметрами используется для инициализации параметров объекта класса значениями “из вне”.

5. В каких случаях вызывается конструктор копирования?

Когда нужно получить полную копию объекта.

6. Перечислить свойства конструкторов.

* Имя класса и конструктора совпадают.
* Ничего не возвращают
* Расположены в public
* Могут быть перегружены.

7. Перечислить свойства деструкторов.

* Имя как у класса со знаком “~” перед.
* Расположен в public
* Ничего не принимает
* Ничего не возвращает
* Может быть только

8. К каким атрибутам имеют доступ методы класса?

Ко всем

9. Что представляет собой указатель this?

Указатель, который хранит адрес объекта, который вызывает метод.

10. Какая разница между методами определенными внутри класса и вне класса?

Особой разницы нет, только в синтаксисе.

11. Какое значение возвращает конструктор?

Никакое

12. Какие методы создаются по умолчанию?

Конструктор копирование, конструктор без параметров, деструктор, присваивание.

13. Какое значение возвращает деструктор?

Никакое

14. Дано описание класса

class student

string name;

int group;

public:

student(string, int);

student(const student&)

~student();

Какой метод отсутствует в описании класса?

Конструктор без параметров

15. Какой метод будет вызван при выполнении следующих операторов:

student\*s;

s=new student;

Конструктор без параметров

16. Какой метод будет вызван при выполнении следующих операторов:

students("Ivanov",20);

Конструктор с параметрами

17. Какие методы будут вызваны при выполнении следующих операторов:

student s1("Иванов",20);

student s2=s1;

Конструктор копирования

18. Какие методы будут вызваны при выполнении следующих операторов:

student s1("Иванов",20);

student s2;

s2=s1;

Конструктор с параметрами и конструктор копирования

19. Какой конструктор будет использоваться при передаче параметра в функцию print()

void print(student a)

{a.show()}

Конструктор без параметров

20. Класс описан следующим образом:

class Student

string name;

int age;

public:

void setname(string);

void set\_age(int );

Student p;

Каким образом можно присвоить новое значение атрибуту name объекта р?

С помощью методов setname, set\_age.