**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Языки программирования

|  |
| --- |
| Лабораторная работа №1. Повторение основ работы с классами. |

Руководитель А.В. Проскурин

подпись, дата инициалы, фамилия

Обучающийся БПИ23-02, 23151451 С.А. Черкашин

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2024 г.

# ЦЕЛЬ РАБОТЫ

*Проверка и получение навыков: использования языковых средств и способов динамического выделения памяти; организации форматного ввода и вывода; Получение практических навыков разработки и отладки программ, использующих объекты класса и файловый ввод/вывод. Изучение различных реализаций конструкторов и областей их применения, а также аспектов перегрузки операций и работы с дружественными функциями.*

# порядок выполнения работы

1. Ознакомиться с общей постановкой задачи.
2. Ознакомится с вариантом задания – соответствует вашему номеру в списке группы.
3. Разработать класс согласно варианту задания.
4. Написать и отладить программу на подготовленных наборах тестовых данных.
5. Подготовить отчет по лабораторной работе. Отчет должен включать в себя:
6. Титульный лист;
7. Цель лабораторной работы;
8. Постановку задачи;
9. Текст программы с комментариями;
10. Демонстрацию работы программы *(Снимки экрана при выполнении действий программы с описанием).*
11. Краткие ответы на контрольные вопросы;
12. Выводы по лабораторной работе.
13. Защитить лабораторную работу перед преподавателем.

# постановка задачи

Необходимо разработать программу, формирующую и обрабатывающий **динамический массив объектов класса**.

В реализуемой программе необходимо предусмотреть возможность **демонстрации работы конструкторов и методов** созданного класса.

|  |  |
| --- | --- |
| **Оценка** | **Общее задание** |
| **Удовлетворительно** | 1. Определить класс (согласно варианту задания).   Все поля должны быть закрытыми (private).  Все методы, кроме статических (static), работают только **с одним** объектом класса (не массивом). |
| 1. Написать конструкторы:  * по умолчанию; * с параметрами; * копирования (получающий параметр «ссылка на класс»). |
| 1. Написать методы ввода-вывода данных. |
| 1. Написать методы get\_**X** для получения значения поля **X**. |
| 1. Написать методы set\_**X** для установки значения поля X. |
| 1. Реализовать выборки (согласно варианту задания). |
| **Хорошо** | 1. Реализовать перегрузку операций для ввода-вывода данных через поток с помощью обычных функций. |
| 1. Написать функцию или статический метод заполнения массива объектов класса данными из файла (имя файла указывает пользователь программы, первой строчкой в файле указано количество записей). В качестве аргумента нужно передать ссылку на открытый поток для чтения (std::istream &s). |
| 1. Написать функцию или статический метод вывода в файл информации (согласно варианту задания) (имя файла указывает пользователь программы). В качестве аргумента нужно передать открытый поток для записи (std::ostream &s). |
| **Отлично** | 1. Написать методы для перегрузки операций сравнения (согласно варианту задания). |
| 1. Выполнить сортировку массива, используя перегрузки операций сравнения (согласно варианту задания). |
| 1. С помощью функций реализовать меню для организации взаимодействия с пользователем.   С помощью меню необходимо выполнить демонстрацию функционирования всей программы (т.е. после выполнения действия происходит вывод меню до тех пор, пока в меню не будет выбран пункт «Закончить работу с программой»). |

**Вариант №24.**

**Класс PersonData** (Персональные данные):

**Поля**: Фамилия, Имя, Отчество, Пол, Образование, Год рождения.

*Реализовать выборки:*

а) списка граждан, возраст которых превышает заданный;

б) списка граждан с высшим образованием, чей возраст находится в определенном диапазоне;

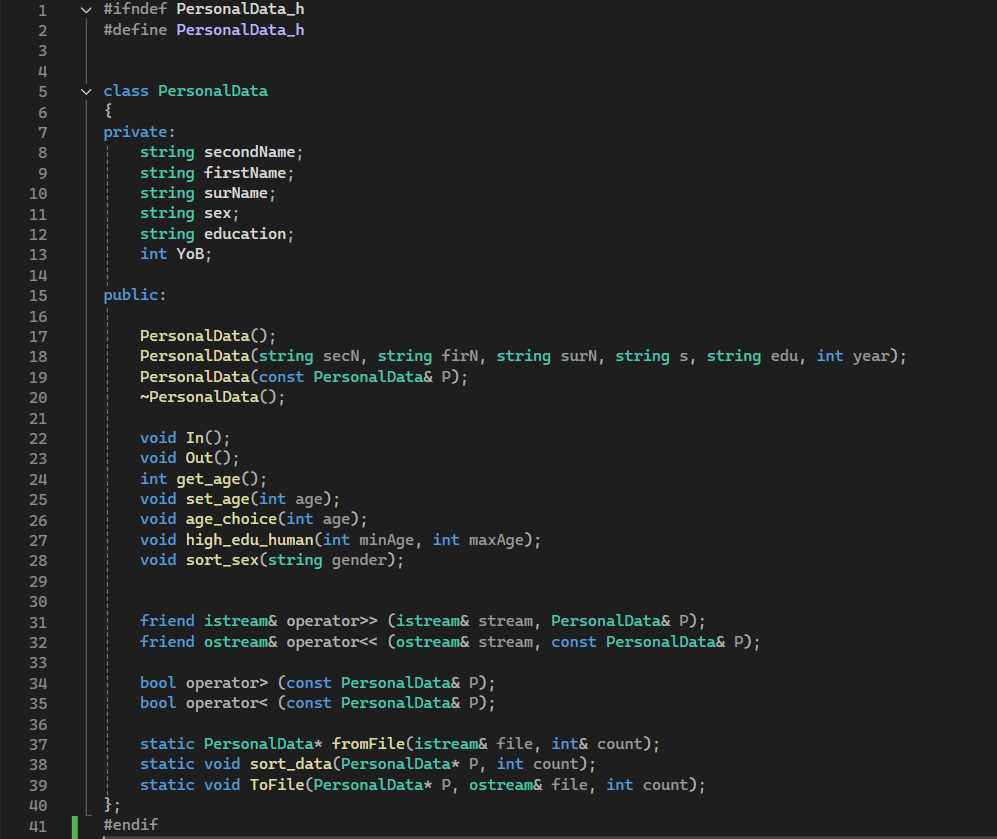
в) списка граждан заданного пола.

*Реализовать сортировку массива объектов класса на основе:*

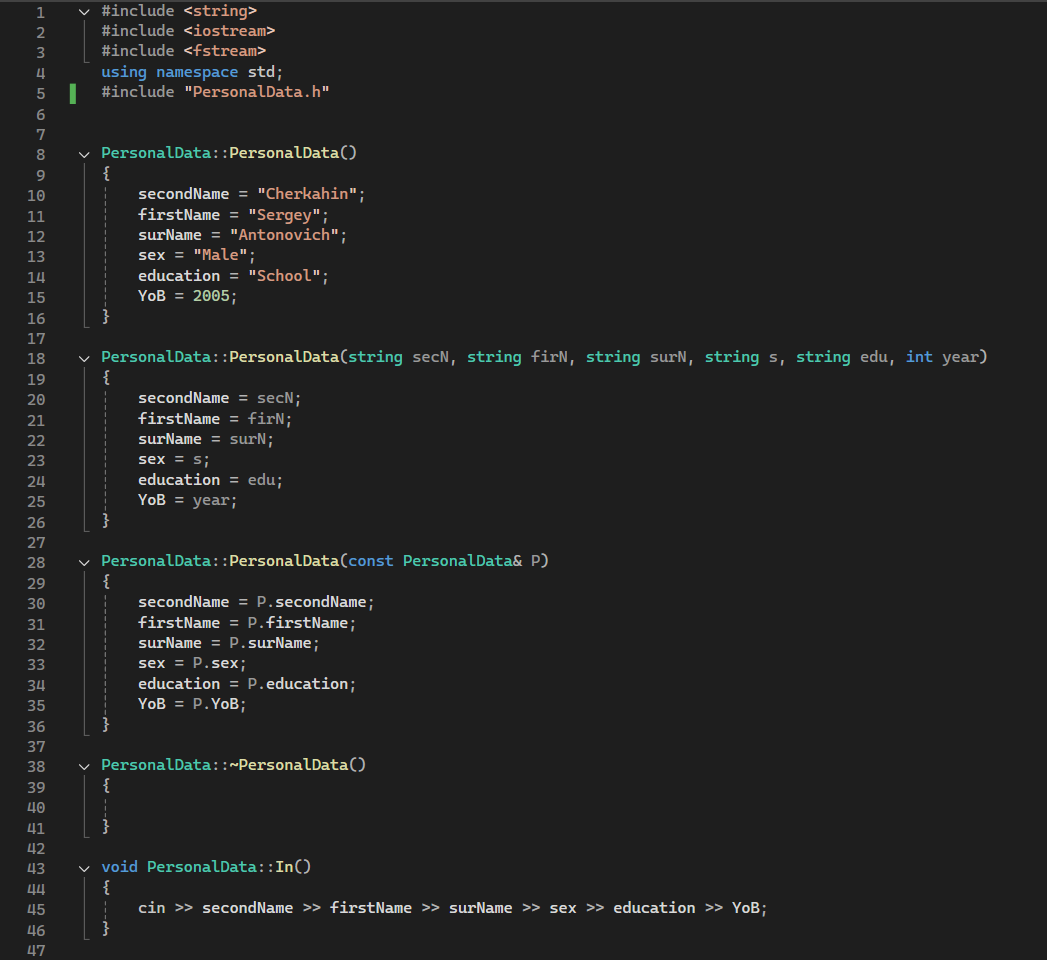
- года рождения.

# ХОД РАБОТЫ

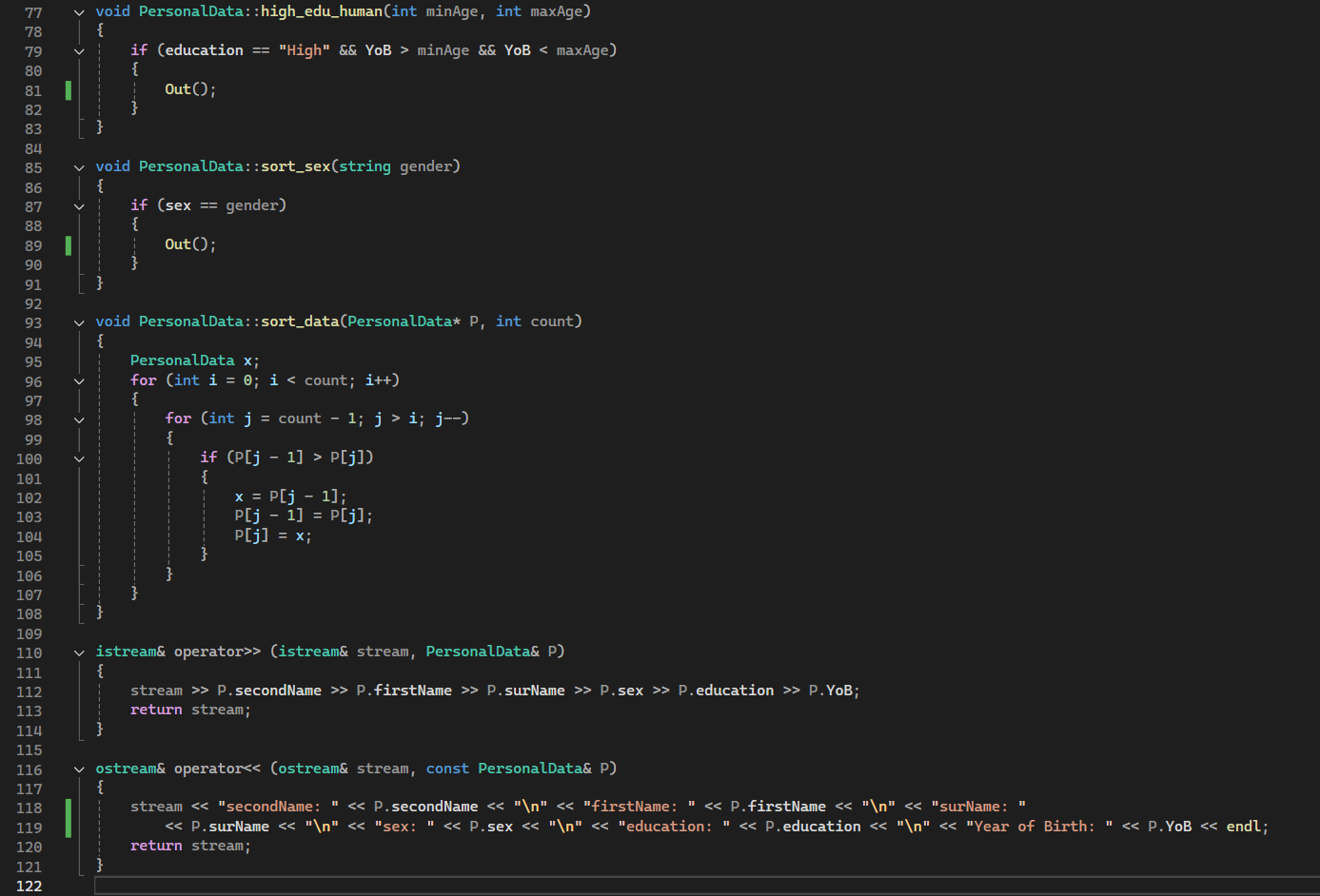
PersonalData.h

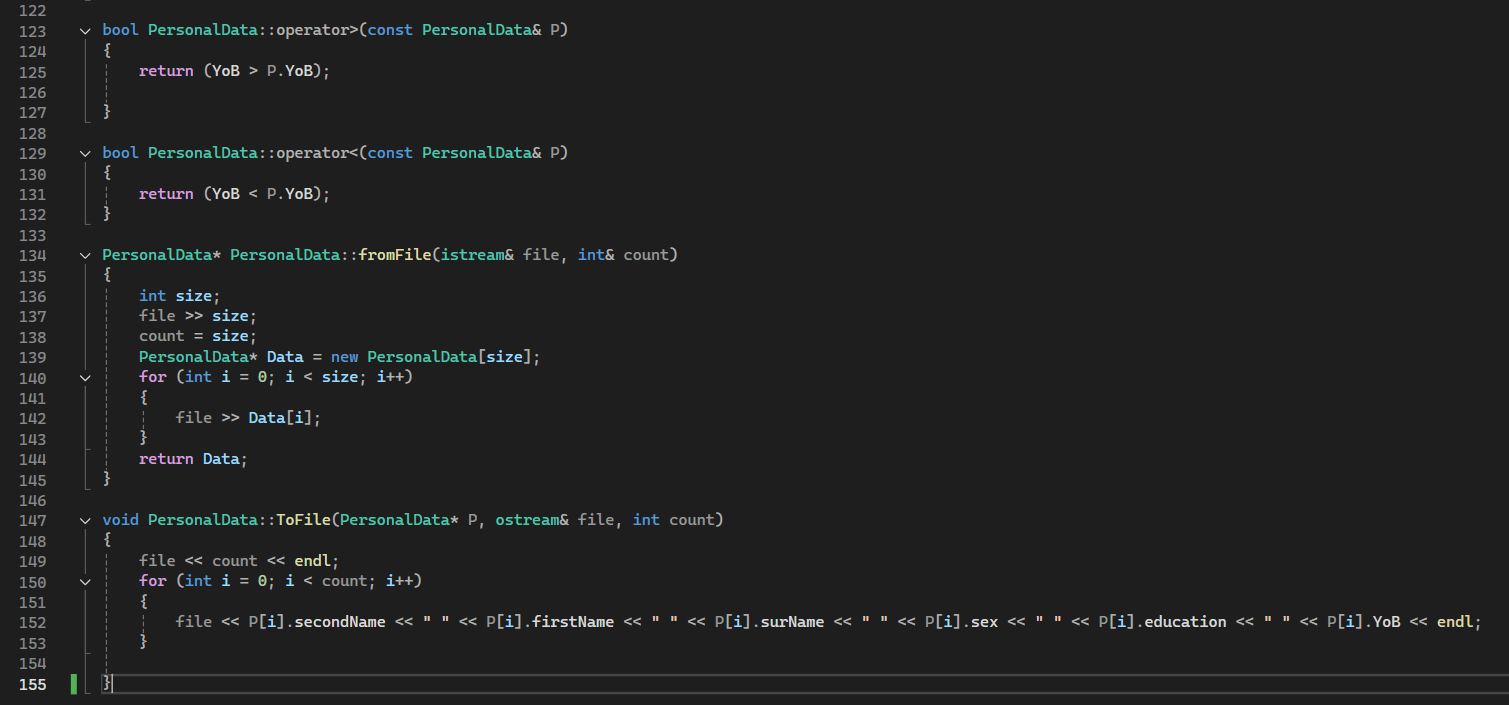


PersonalData.cpp

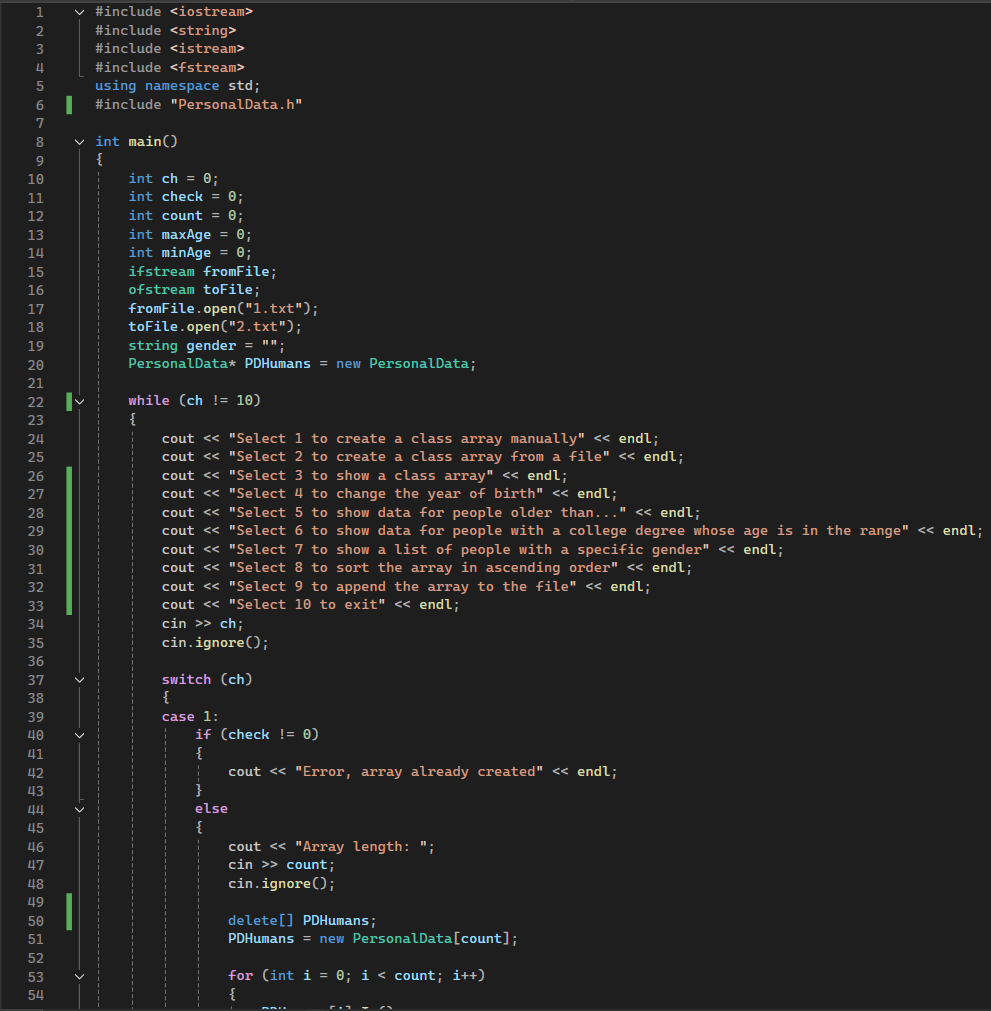


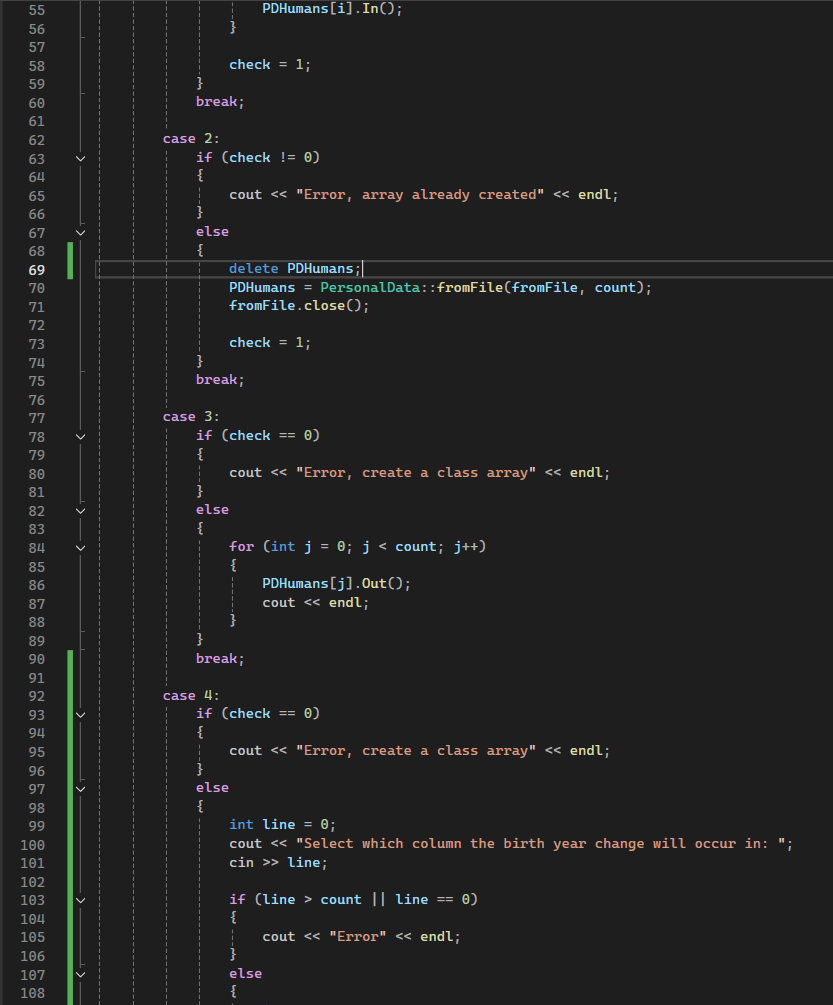


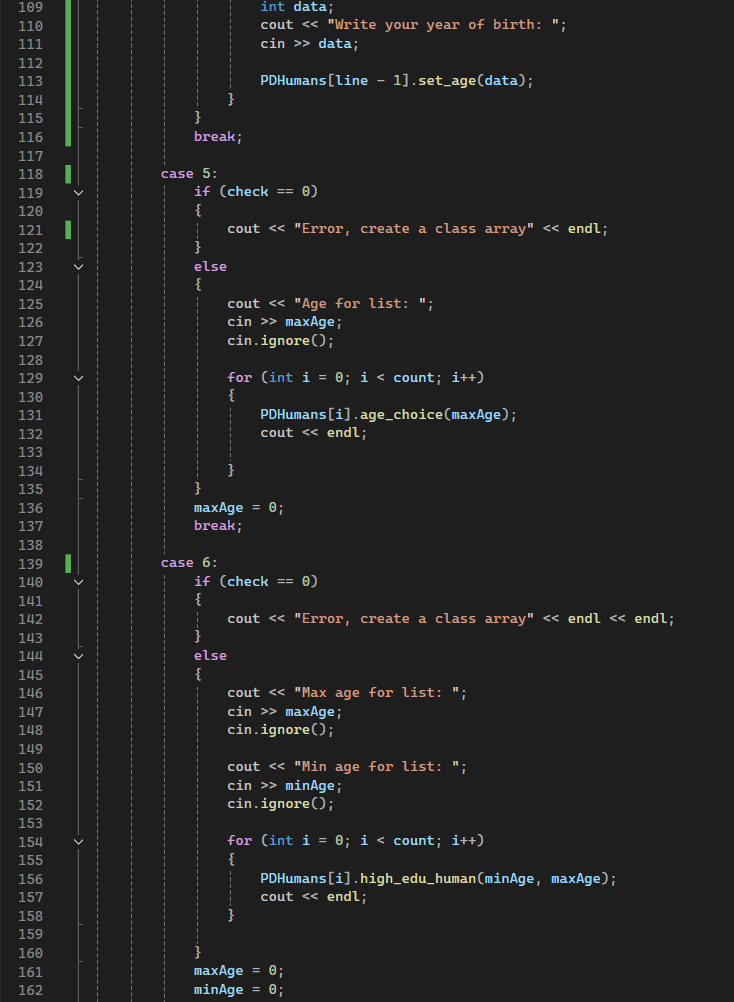


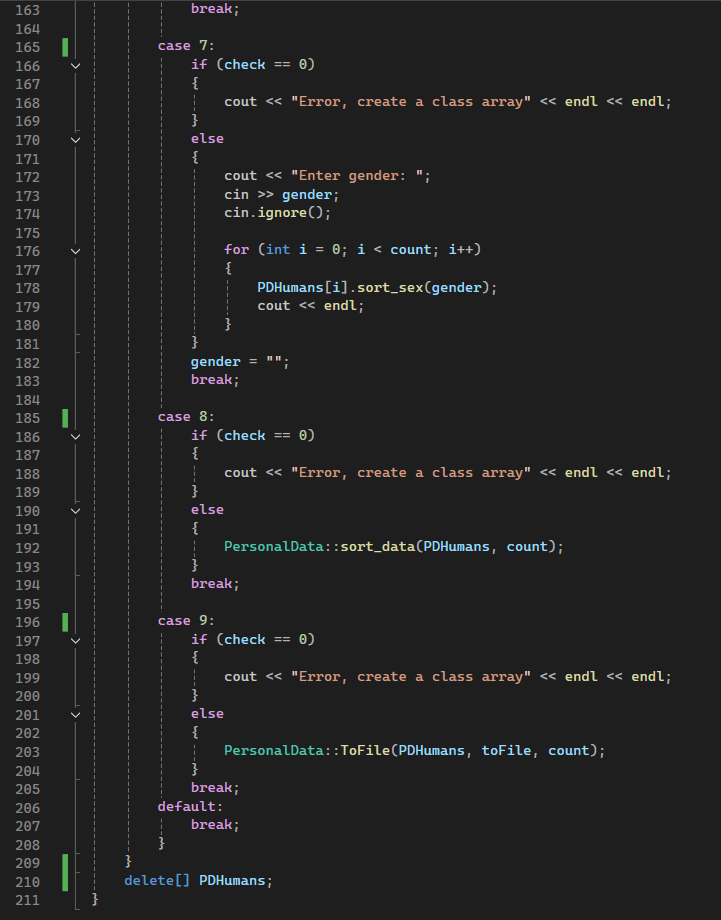


Main.cpp

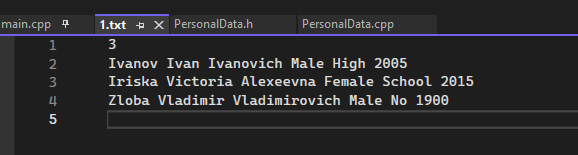




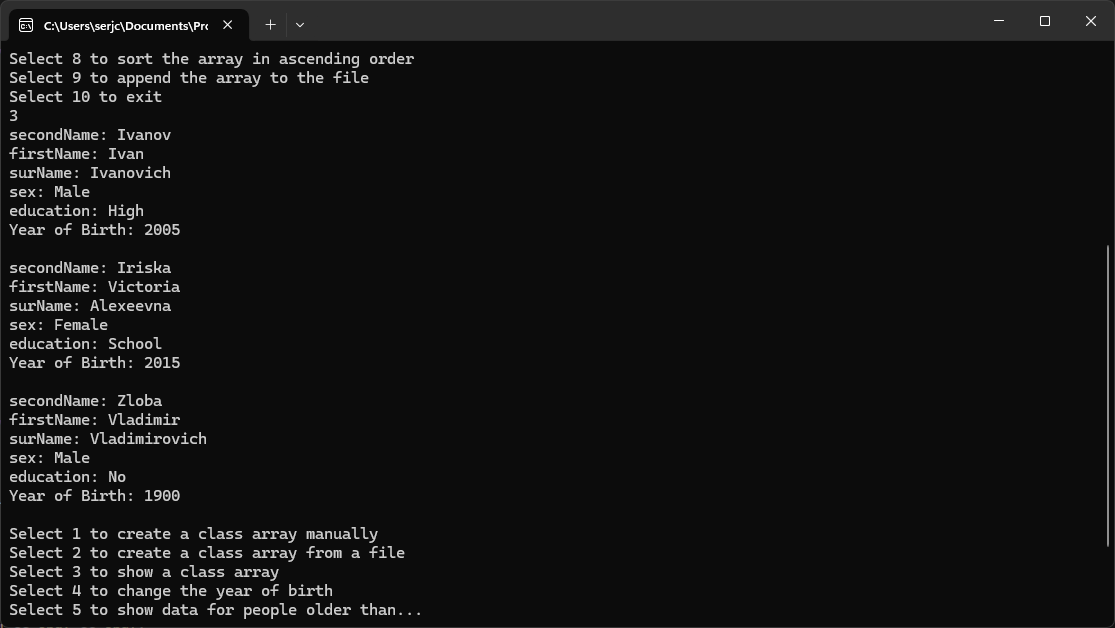




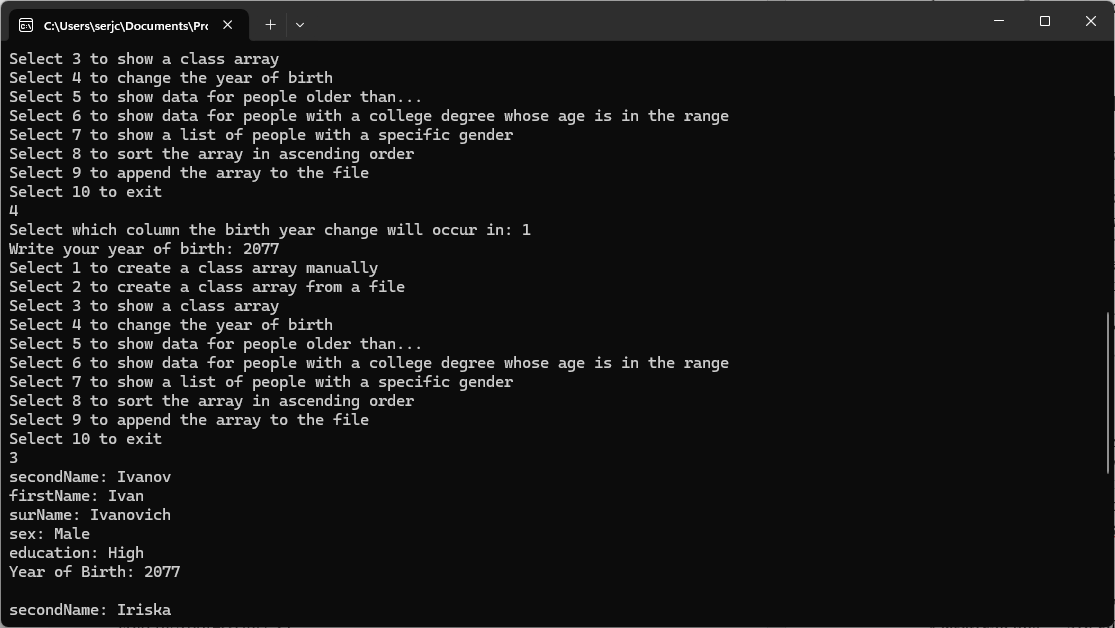
Примеры работы:



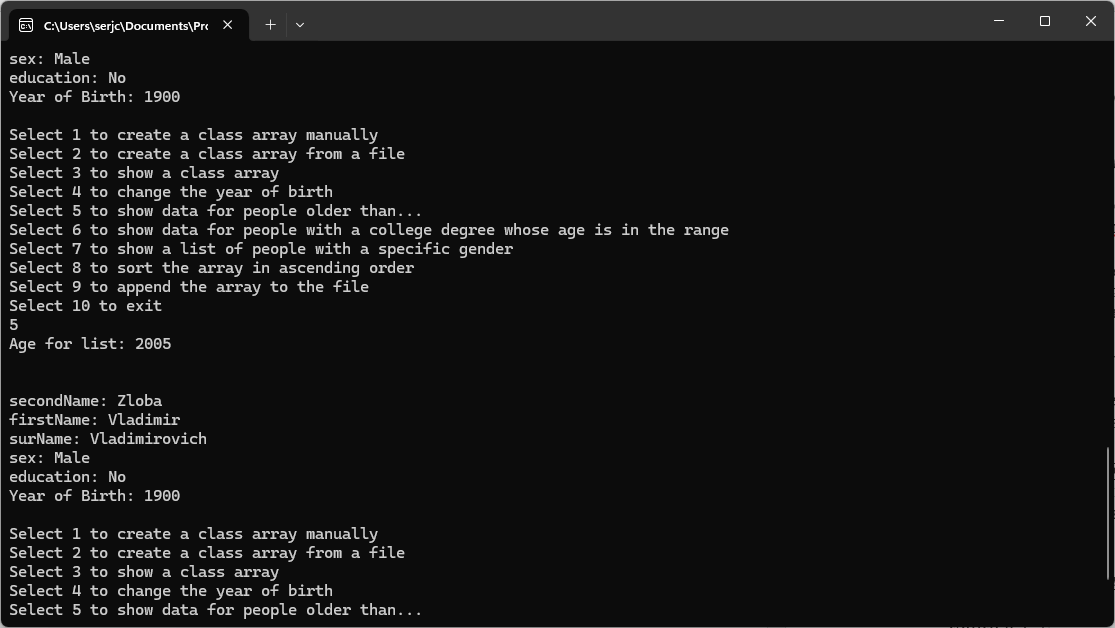
Данные из файла



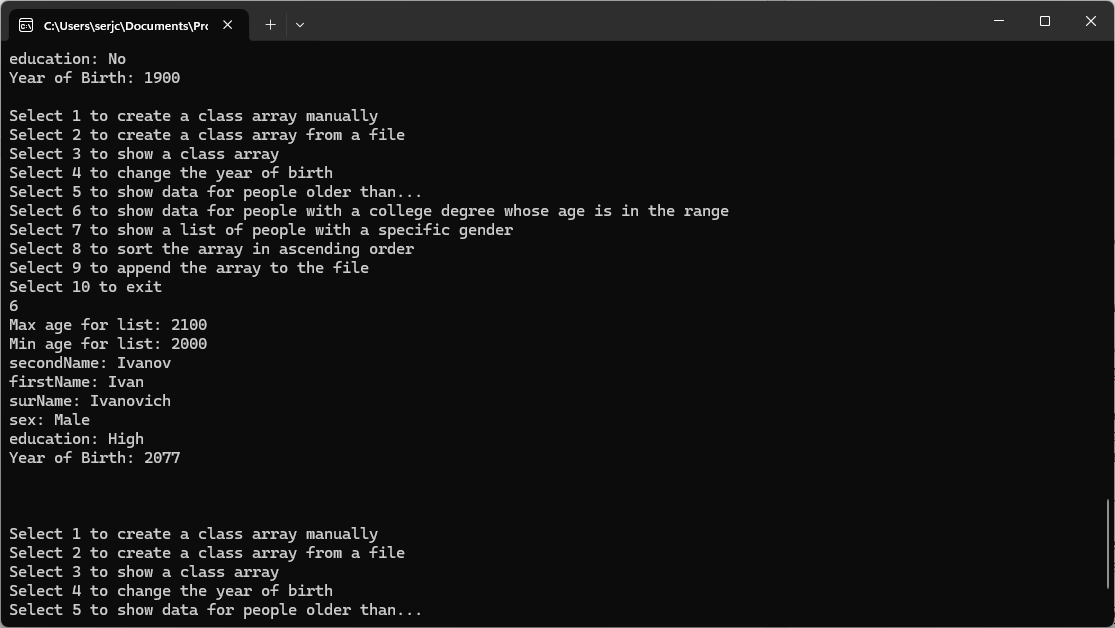
Изменение даты рождения:



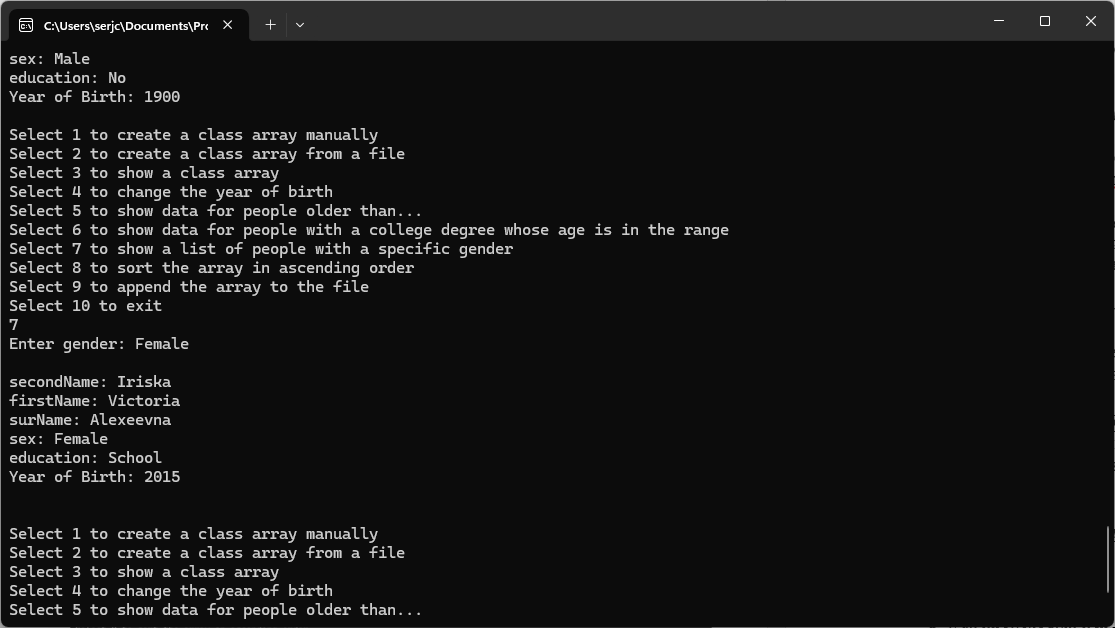
Показать список людей, которые старше определенной даты:



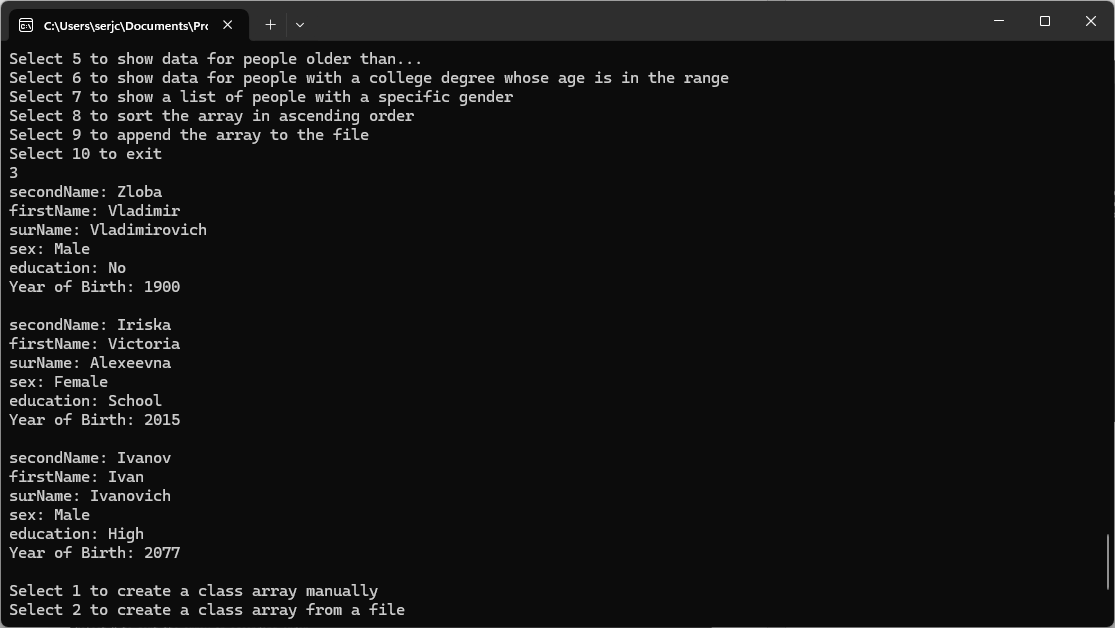
Показать список людей, у которых высшее образование и датой рождения от... и до…



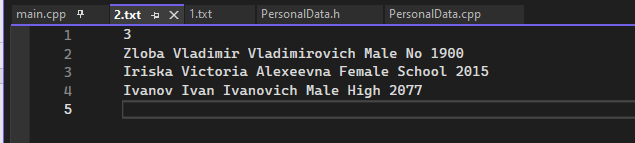
Сортировка по гендеру:



Сортировка по дате рождения по возрастанию



Запись в файл:



# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. *Раскройте понятие указателя в языке С. Опишите синтаксис объявления указателя.*

**Указатель** — это переменная, которая хранит адрес другой переменной.

Синтаксис указателя: int \*ptr;

1. *В чем преимущество использования операций new и delete в языке С++ по сравнению с аналогичными средствами языка С?* 
   1. Автоматическая инициализация объектов при использовании **new**
   2. Возможность вызова деструктора при удалении объекта через **delete.**
2. *В чем заключаются преимущества использования динамических массивов?*

Изменять размер во время выполнения программы, а также эффективно использовать память, так как массивы выделяются только при необходимости.

1. *Как производится обращение к элементам динамических массивов?*

Для обращения к элементам динамического массива используется указатель, выделенный через new.

*int\* arr = new int[size];*

*arr[0] = 10;*

1. *Как производится определение пользовательских функций в языке С?*

int Func(int a) {

return a \* 2;

}

1. *Как происходит передача параметров в функцию по значению/ адресу?*

**По значению**: передается копия аргумента, изменения не затрагивают исходный объект.

**По адресу**: передается адрес аргумента, изменения влияют на исходный объект.

void byValue(int x);

void byAddress(int\* x);

1. *Какого типа значения может возвращать функция? Как вернуть из функции несколько значений?*

Функция может возвращать значения типа bool, int, float, char, str. Для возврата нескольких значений можно использовать структуру.

1. *Как организуется доступ к файлам? Какие виды доступа вы знаете? Каким образом можно считать/записать данные из текстового файла?*

Виды доступа: чтение, запись, добавление

Работа с файлом происходит через потоки.

fin >> chislo - ввод из файла

fin << chislo – вывод из файла

1. *Что такое класс? Как выполнить объявление класса?*

**Класс** — это пользовательский тип данных, объединяющий данные и функции. Объявление класса:

class MyClass {

int data;

public:

void setData(int value);

};

1. *Как в классе реализуется инкапсуляция?*

**Инкапсуляция** скрывает внутреннюю реализацию класса, предоставляя доступ к данным только через методы.

1. *Каким образом элементы класса получают атрибуты доступа?*

Атрибуты класса могут иметь модификаторы доступа: **public**, **private**, **protected**, которые определяют уровень доступа. При создании класса, мы определяем, какой доступ получит элемент класса.

1. *Что такое объект класса и как выполнить объявление объекта класса в программе?*

Объект класса – это экземпляр класса, или его копия, которая находится в памяти компьютера

Например, можно объявить объекты класса Box:

Box d; // создаётся объект класса Box

Box \* dptr = &d; // создаётся указатель на объект d

Box & dref = d; // создаётся ссылка на объект d

1. *Что такое методы класса? Каковы особенности объявления и определения методов класса?*

**Методы** — это функции, определенные внутри класса. Они имеют доступ к данным класса и объявляются так:

void MyClass::setData(int value) {

data = value;

}

1. *Что такое указатель* ***this****? Как выполняется вызов метода класса?*

Указатель this — это специальный указатель, который автоматически передаётся всем нестатическим методам класса. Он содержит адрес текущего объекта, для которого был вызван метод. Таким образом, this указывает на сам объект, на котором вызывается метод.

Вызов метода класса выполняется следующим образом:

**Через объект класса:**

MyClass obj;

obj.setValue(10)

Через указатель на объект:

MyClass\* ptr = &obj;

ptr->setValue(20)

1. *Какие функции называются конструкторами? Каков механизм передачи параметров конструктору?*

**Конструктор** – это метод создания и инициализации объектов данного класса.

Уникальные свойства конструкторов:

* конструкторы имеют то же имя, что и класс;
* не возвращают значение и не имеют тип возвращаемого значения;
* не могут быть унаследованы, но производный класс (наследник) может вызывать конструкторы базового класса (родителя);
* конструктор вызывается неявно при создании или копировании объекта данного класса. Вызвать явно его нельзя.

1. *Что такое конструктор по умолчанию и какую пользу приносит его наличие?*

Если конструктор без параметров в классе явно не определен, то будет использоваться конструктор по умолчанию, созданный компилятором.

Конструктор по умолчанию вашего класса вызывает конструкторы по умолчанию для каждого поля этого класса.

Если конструктор явно определен в классе, то конструктор по умолчанию не будет создаваться автоматически.

1. *Как выполняется инициализация объектов класса?*

Через различные конструкторы: по умолчанию, копии, с параметрами.

1. *Каким образом в программе можно создать объект класса, элементы-данные которого являются копиями элементов данных созданного ранее объекта?*

С помощью конструктора по умолчанию.

1. *Каким образом осуществляется перегрузка операций в языке С++?*

Для **перегрузки** встроенной **операции** (+, -, != и т.д.) используется ключевое слово operator, после которого следует знак этой операции.

1. *Какие операции не могут быть перегружены?*

Перегружать нельзя следующие операции:

* ?: (тернарный условный оператор);
* :: (доступ к вложенным именам);
* . (доступ к полям);
* sizeof (нахождения размера в байтах);
* typeid (определение типа переменной).

1. *Теряет ли операция при перегрузке свое исходное назначение?*

Когда **оператор перегружается**, ни одно из его первоначальных **назначений не теряется**

1. *Каков синтаксис объявления оператора-функции?*

тип\_возвр\_зн operator знак\_операции (список\_параметров)

1. *Почему при перегрузке унарных операций оператор-функция не имеет параметров?*

При перегрузке унарных операций оператор-функция не имеет параметров, потому что эти операции выполняются только над одним объектом — объектом, для которого вызывается этот оператор.

1. *Как различить префиксную и постфиксную операции инкремента при перегрузке?*

При постфиксной форме, когда задается перегрузка операции в списке параметров указывается тип данных int

1. *Какая функция называется другом класса?*

**Дружественная функция** – это функция, не являющаяся методом класса, но имеющая доступ к его закрытым (private) и защищенным (protected) элементам.

1. *В чем состоят различия между дружественной функцией и методом класса?*

* Объявляется внутри класса с модификатором friend
* Не является элементом класса, поэтому ей не передаётся указатель this. Объект класса нужно передавать явно через параметр.

1. *В чем состоят особенности передачи параметров дружественной функции при перегрузке унарных и бинарных операций?*

При перегрузке унарных операций через дружественную функцию унарный оператор работает только с одним объектом. Этот объект необходимо передавать в качестве параметра.

Для бинарных операций, таких как +, -, \*, и других, при перегрузке через дружественную функцию необходимо передать оба операнда явно, поскольку дружественная функция не имеет доступа к объекту через this.

1. *Почему нельзя перегрузить операции>> и <<для ввода и вывода типов с помощью метода класса?*

Поскольку на первом месте в параметрах обязательно должен быть поток, то перегрузку операций>> и <<можно сделать только функцией.

# ВЫВОДЫ

Было вспомнено и повторено использование классов, а также создание статических методов. Было повторено использование перегрузок, а также повторены тонкости работы с файлами.