Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе**

Дисциплина: Основы алгоритмизации и программирования

Тема: Сохранение данных в файле с использованием потоков

Выполнил работу

студент группы РИС-22-1б

Рыжков Н.С.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2023

**Описание задачи**

Необходимо создать класс `marsrut`, который будет содержать информацию о маршруте. Класс должен иметь следующие методы:

- Конструктор без параметров

- Конструктор с параметрами

- Конструктор копирования

- Деструктор

- Методы доступа к полям класса (getters и setters)

- Метод `show()`, который будет выводить информацию о маршруте на экран

- Функцию `make\_marsrut()`, которая будет запрашивать информацию о маршруте у пользователя и создавать объект класса `marsrut`

- Функцию `print\_marsrut()`, которая будет выводить информацию о маршруте на экран

**Список методов класса marsrut**

Конструктор без параметров

marsrut();

Создает объект класса `marsrut` без параметров.

Конструктор с параметрами

marsrut(string, string, float);

Создает объект класса `marsrut` с заданными параметрами:

- `string` `firstplace` - название первого пункта маршрута

- `string` `secondplace` - название второго пункта маршрута

- `float` `rasstoyanie` - расстояние между пунктами маршрута в километрах

Конструктор копирования

marsrut(const marsrut&);

Создает копию объекта класса `marsrut`.

Деструктор

~marsrut();

Удаляет объект класса `marsrut`.

**Методы доступа к полям класса**

Getter для поля `firstplace`

string get\_firstplace();

Возвращает название первого пункта маршрута.

Setter для поля `firstplace`

void set\_firstplace(string);

Устанавливает название первого пункта маршрута.

Getter для поля `secondplace`

string get\_secondplace();

Возвращает название второго пункта маршрута.

Setter для поля `secondplace`

void set\_secondplace(string);

Устанавливает название второго пункта маршрута.

Getter для поля `rasstoyanie`

float get\_rasstoyanie();

Возвращает расстояние между пунктами маршрута в километрах.

Setter для поля `rasstoyanie`

void set\_rasstoyanie(float);

Устанавливает расстояние между пунктами маршрута в километрах.

Метод `show()`

void show();

Выводит информацию о маршруте на экран в формате:

firstplace: <название первого пункта маршрута>

secondplace: <название второго пункта маршрута>

rasstoyanie: <расстояние между пунктами маршрута в километрах>

Функция `make\_marsrut()`

marsrut make\_marsrut();

Запрашивает у пользователя информацию о маршруте и создает объект класса `marsrut` с заданными параметрами.

Функция `print\_marsrut()`

void print\_marsrut(marsrut);

Выводит информацию о маршруте на экран с помощью метода `show()`.

**Заключение**

Класс `marsrut` реализует функционал для работы с маршрутами. Он позволяет создавать, изменять и выводить информацию о маршрутах. Класс можно использовать в приложениях, связанных с транспортом или путешествиями.

