

Статические члены класса.
Обработчик исключений try-
catch

Статические члены класса

Что такое статическая переменная класса?

— такая переменная класса, которая будет существовать независимо от всех объектов этого класса.

Такая переменная используется в контексте всего класса, а не конкретного объекта.

Объявление:

```
static <тип> <имя_статической_переменной>;
```

Инициализация:

```
<тип> <имя_класса>::<имя_статической_переменной> = <выражение>;
```

Что такое статический метод класса?

— метод класса, который можно вызывать без создания объекта.

Такой метод используется в контексте всего класса, а не конкретного объекта.

```
class Math {  
public:  
    static float add(float a, float b) {  
        return a + b;  
    }  
    static int int_divide(int a, int b) {  
        return int (a / b);  
    }  
}
```

```
};  
  
int main() {  
    Math math;  
    cout << math.add(5, 999.1);  
    cout << < Math::int_divide(13, 2);  
    return 0;  
}
```

Task 2.1

1. Написать статический метод `Register`, который регистрирует нового пользователя, записывая в свой файл информацию о последнем выданном `id`, и создаёт новый файл с информацией о новом пользователе

Обработчик исключений try-catch

ИСКЛЮЧИТЕЛЬНАЯ СИТУАЦИЯ

— любое поведение программы, которое будет названо так программистом.

Например, деление на 0

```
class Math {  
public:  
    static float add(float a, float b) {  
        return a + b;  
    }  
    static int int_divide(int a, int b) {  
        if (b == 0) throw "Error: Zero division";  
        return int(a / b);  
    }  
};
```

```
int main() {  
    Math math;  
    try {  
        cout << math.add(5, 999.1) << endl;  
        cout << Math::int_divide(13, 0) << endl;  
    }  
    catch (const char* message) {  
        cout << message << endl;  
    }  
    return 0;  
}
```


Who is who

- **throw** – «выбрасывает» исключение; можно «выбросить» любой объект (переменную)
- **try** – пытается выполнить код; если в процессе выполнения было «брошено» исключение, передает выполнение в `catch`
- **catch** - «ловушка» исключения;
 - `catch(Type)` – «ловит» все исключения типа `Type`
 - `catch(Type variable)` – «ловит» все исключения типа `Type`, при этом можно обработать переменную `variable` (например, вывести сообщение)
 - `catch(...)` – «ловит» исключения всех типов

Task 2.2

1. Добавить обработку исключений в случае ошибки открытия файла («File can't be found»)

Необходимо знать

1. Статические переменные и методы
2. Обработчик исключений

Полезные книжки

1. С.Прата – Язык программирования C++. Лекции и упражнения (6+ издание)
2. Б.Страуструп – Язык программирования C++ (3+ издание)
3. Г.Шилдт – Самоучитель C++ (3+ издание)