Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа 9

**Обработка исключительных ситуаций**

Выполнил

Студент гр. РИС-24-3б Рунтов Иван Андреевич

Проверил

Доцент кафедры ИТАС Ольга Андреевна Полякова

г. Пермь, 2025

**Постановка задачи**

1. Реализовать класс, перегрузить для него операции, указанные в варианте.

2. Определить исключительные ситуации.

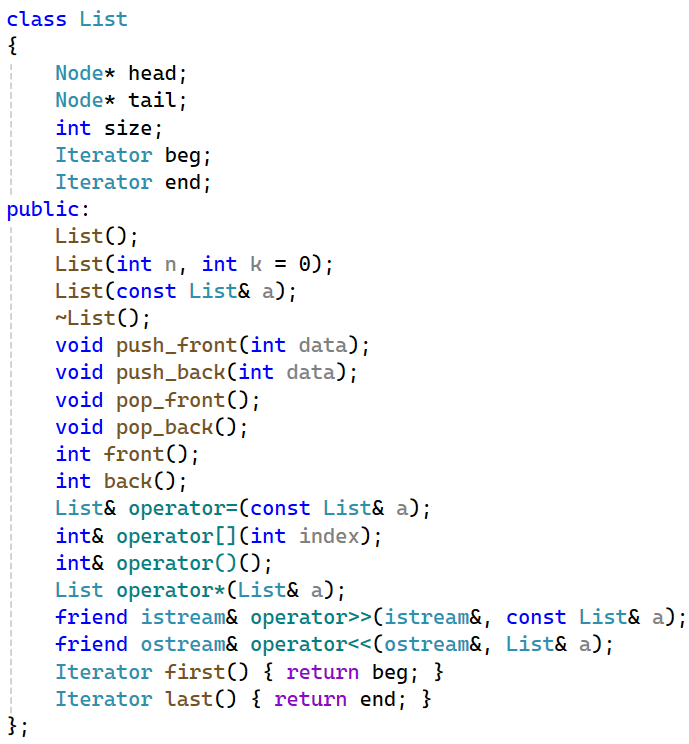
3. Предусмотреть генерацию исключительных ситуаций.

**Вариант 15**

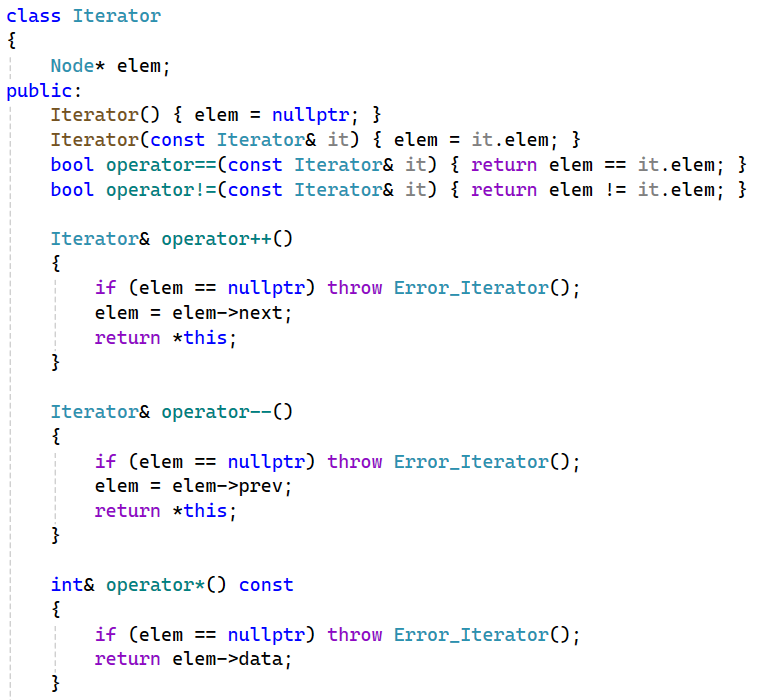
Класс- контейнер СПИСОК с ключевыми значениями типа int.

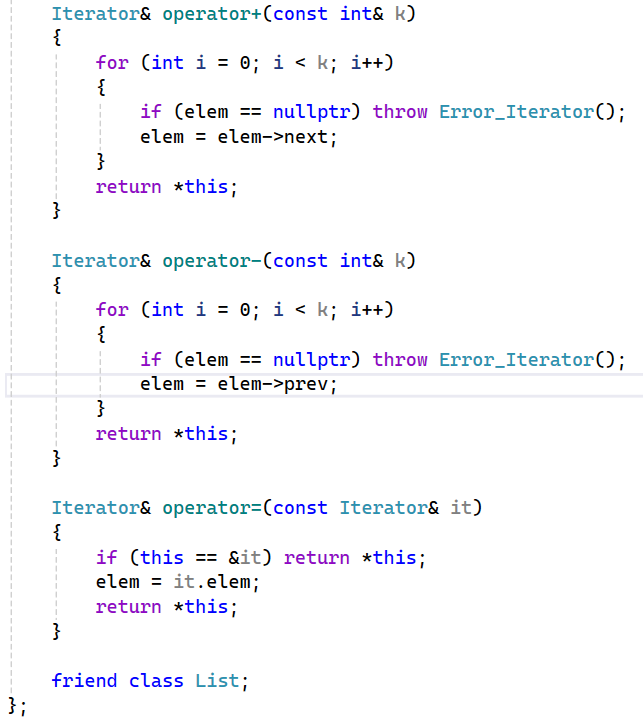
Реализовать операции: [] – доступа по индексу; int() – определение размера списка; \* вектор – умножение элементов списков a[i]\*b[i]; +n - переход вправо к элементу с номером n.

**Описание класса List**



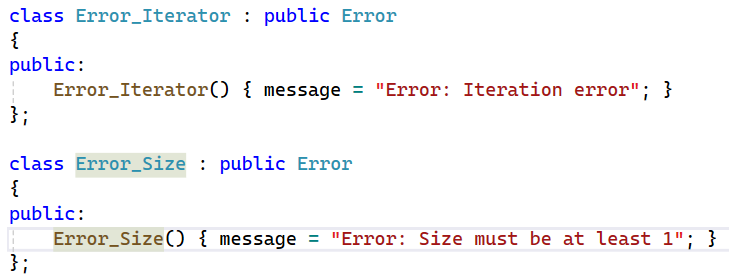
**Описание и определение класса Iterator**





**Определение и описание класса Error и его производных классов**





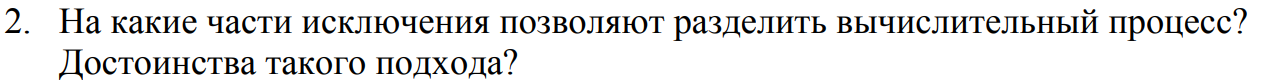
**Функция main**



**Ответы на контрольные вопросы**



Исключение в C++ - объект, который генерируется в непредвиденной ситуации.



Процесс разделяется на 2 части:

1. Обнаружение аварийной ситуации
2. Обработка аварийной ситуации

Преимущества:

1. Удобно использовать в программе, состоящей из нескольких модулей
2. Не нужно возвращать значение в функцию



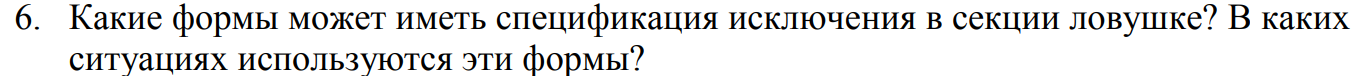
Оператор throw



Контролируемый блок try является блоком, в котором может быть сгенерировано исключение. В него помещается тот код, где может случиться аварийная ситуация.



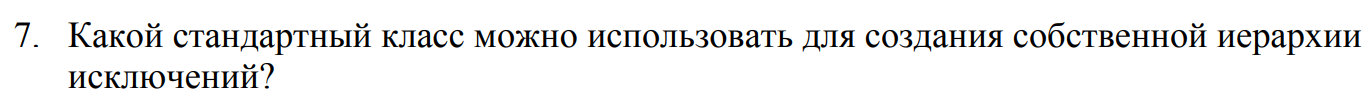
Секция-ловушка catch необходима для перехвата и обработки исключения конкретного типа.



catch(тип i) – в данном случаем объект передаются в ловушку и может быть использован в ней

catch(тип) – в данном случае объект не передается в ловушку и не может использоваться внутри (важно знать только тип)

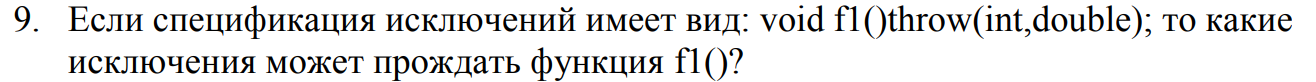
catch(…) – ловушка перехватывает все исключения. Прописывается полсле всех ловушек для обработки неперехваченных исключений.



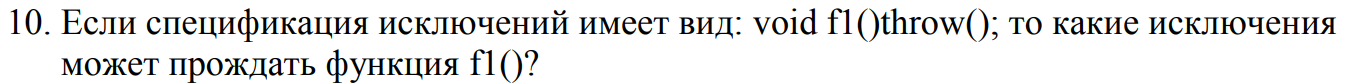
Для создания собственной иерархии исключений можно использовать стандартный класс exception



Необходимо создать собственный базовый класс-исключение и унаследовать от него классы, отвечающие за разные исключения.



В данном случае функция f1() может порождать только исключения типа int и double

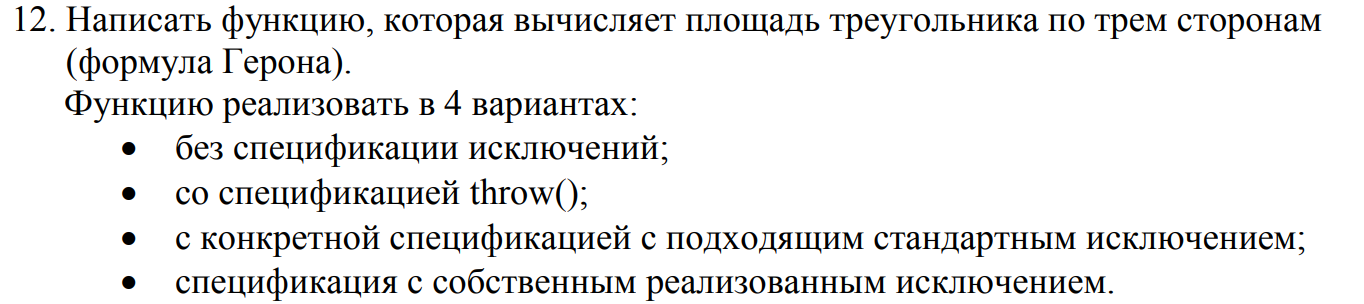


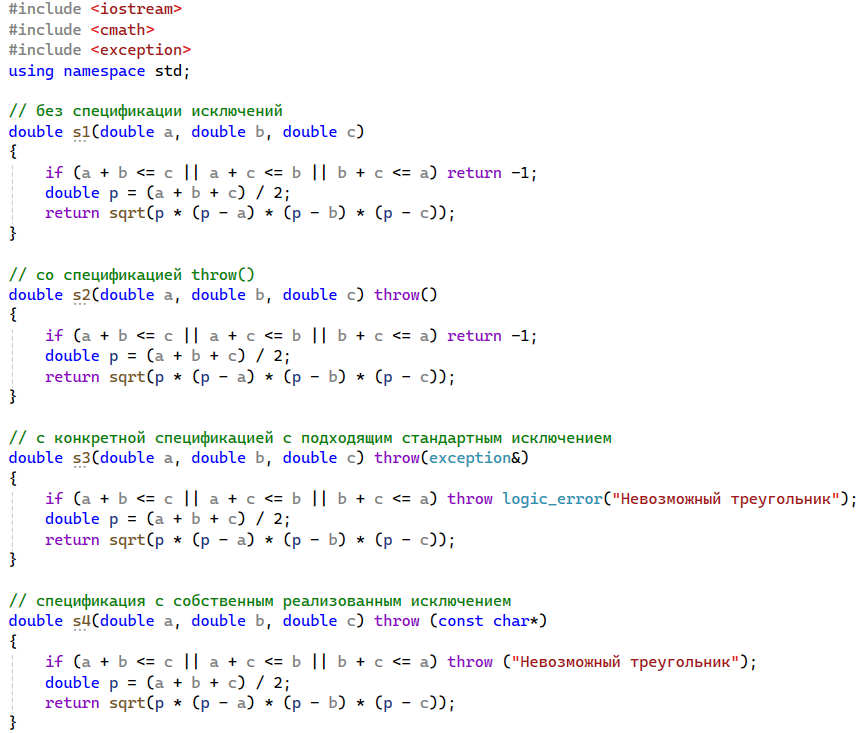
В данном случае функция f1() не может порождать никакие исключения

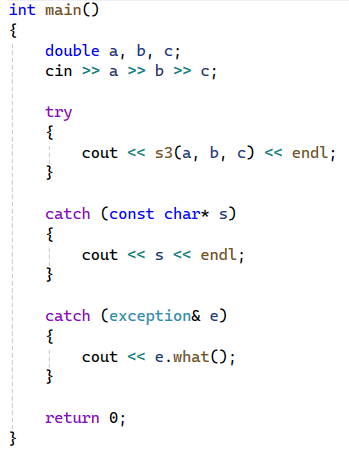


Синхронные исключения – такие исключения могут возникнуть только в определенных точках программы.

Асинхронные исключения – могут возникать в любой момент и независимо от того, что выполняет программа







Ссылка на github: <https://github.com/RuntovIvan/Informatika>