

6. Лабораторная работа № 6. Исключения

Варианты заданий для лабораторной работы № 6

Номер варианта выбирается следующим образом: № % 9, где № - номер студента в списке группы ЭУ, отсортированной по алфавиту, если результат равен нулю, то выбирается вариант 9.

Вариант 1

Реализовать класс «Время». Поля: часы, минуты, секунды. Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение. Для исключения создать класс, производный от `out_of_range`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Время» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 2

Реализовать класс «Дата». Поля: число, месяц, год. Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение. Для исключения создать класс, производный от `out_of_range`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Дата» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 3

Реализовать класс «Точка в пространстве внутри единичного куба». Поля: координаты точки x , y , z , значения всех полей находятся в интервале $[0, 1]$. Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение. Для исключения создать класс, производный от `out_of_range`. Дополнительно созданный класс для исключения кроме

наследуемых полей (необходимо их также использовать) содержит поля класса «Точка в пространстве внутри единичного куба» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 4

Реализовать класс «Простое число». Поля: значение числа. Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. При недопустимых параметрах объекта (число непростое) создается исключение. Для исключения создать класс, производный от `invalid_argument`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Простое число» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 5

Реализовать класс «Массив – последовательность Фибоначчи». Поля: указатель на массив и его размерность. Конструкторы: конструктор для инициализации полей (через параметры передается обычный массив). При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение (массив не является последовательностью Фибоначчи). Для исключения создать класс, производный от `invalid_argument`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Массив – последовательность Фибоначчи» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 6

Реализовать класс «Массив – арифметическая прогрессия». Поля: первый элемент, разность, указатель на массив, в котором хранятся первые элементы прогрессии и его размерность. Конструкторы: конструктор для инициализации полей (массив — прогрессия передается через параметры). При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение (массив не является заданной арифметической прогрессией). Для исключения создать класс, производный от

`invalid_argument`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Массив – арифметическая прогрессия» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 7

Реализовать класс «Массив – геометрическая прогрессия». Поля: первый элемент, знаменатель, указатель на массив, в котором хранятся первые элементы последовательности и его размерность. Конструкторы: конструктор для инициализации полей (массив — прогрессия передается через параметры). При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение (массив не является заданной геометрической прогрессией). Для исключения создать класс, производный от `invalid_argument`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Массив – геометрическая прогрессия» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 8

Реализовать класс «Квадратное уравнение». Поля: коэффициенты a , b , c . Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. Если квадратное уравнение не имеет действительных корней, то создается исключение. Для исключения создать класс, производный от `invalid_argument`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Квадратное уравнение» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Вариант 9

Реализовать класс «Треугольник». Поля: длины сторон a , b , c . Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. Если с заданными длинами сторон треугольник не существует, то создается исключение. Для исключения создать класс, производный от `invalid_argument`. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать)

содержит поля класса «Треугольник» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.