

Контрольные вопросы:

- ☐ (5 б.) Какие существуют способы обработки различных ошибок?
- ☐ (5 б.) В чем заключаются недостатки механизма кодов возврата?
- ☐ (5 б.) Какими особенностями обладает механизм исключений?
- ☐ (5 б.) Для чего используется спецификатор и оператор noexcept?
- ☐ (5 б.) Как формулируются гарантии безопасности исключений?

Упражнения:

- ☐ (25 б.) Предложите вариант реализации системы обработки ошибок на основе кодов возврата для произвольной функции без использования дополнительного аргумента. Рассмотрите возможность использования структуры и объединения. Набор кодов возврата реализуйте на основе перечисления с областью видимости.
- ☐ (25 б.) Продемонстрируйте использование стандартных исключений на примере простых тестовых ситуаций.
- ☐ (25 б.) Добавьте в класс дроби из предыдущего задания систему обработки ошибок на основе исключений.
- ☐ (25 б.) Каким максимальным количеством путей выполнения может порадовать нас приведенный ниже код?

```
String evaluate_salary_and_return_name(Employee e)
{
    if (e.title() == "CEO" || e.salary() > 100000)
    {
        std::cout << e.name() << " " << e.surname() << " is overpaid.\n";
    }
    else
    {
        std::cout << e.name() << " is not overpaid.\n";
    }

    return e.name() + " " + e.surname();
}
```

Примечание: необходимо указать пути нормального выполнения, а также пути выполнения с исключениями.