

# Общие положения

## *Общие положения Практикума по программированию.*

За семестр будет 4 задания, каждое оценивается в 10 баллов. За просрочку сдачи (сдача после занятия-дедлайна) штраф -2 балла за каждое просроченное занятие, но максимальный штраф за просрочку -6 баллов (т.е. за задание студент получит 4 балла).

Списывание карается штрафом. Тотальное списывание – 0 баллов за задачу без права пересдачи. Локальная неспособность объяснить происходящее тоже штрафуются. Неполный функционал задачи при отсутствии желания доделывать – не полный балл за задачу.

Шкала:

35 баллов и выше – автомат без участия в зачете;

30-34 балла – зачет в виде собеседования (неформальный автомат);

Остальные приходят на зачет с целью получить итоговую сумму не менее 51 балла. На зачете будут решать задачи, которые выдавались в семестре, но с нуля и чужие варианты, при этом балльная оценка задач существенно возрастет (ориентировочно полная задача – 30 баллов).

---

## Задание 2

**Задачи для второго задания Практикума по программированию. Общая тема задания «текстовый калькулятор».**

Базовая часть (выполняется всеми самостоятельно!):

Написать калькулятор для строковых выражений вида '<число> <операция> <число>', где <число> - не отрицательное целое число меньше 100, записанное словами, например "тридцать четыре", <арифметическая операция> - одна из операций "плюс", "минус", "умножить". Результат выполнения операции вернуть в виде текстового представления числа. Пример `calc("двадцать пять плюс тринадцать")` -> "тридцать восемь"

Оформить калькулятор в виде функции, которая принимает на вход строку и возвращает строку.

### Дополнительные задания:

- 1) Реализовать поддержку операции деления и остатка от деления и работу с дробными числами (десятичными дробями). Пример: `calc("сорок один и тридцать одна сотая разделить на семнадцать")` -> "два и сорок три сотых". Обращать дробную часть до тысячных включительно, если при делении получаются числа с меньшей дробной частью выполнять округление до тысячных.

#### **Сложность 2**

- 2) *Расширение задания 1.* Реализовать поддержку десятичной дробной части до миллионных долей включительно. Реализовать корректный вывод информации о периодической десятичной дроби (период дроби вплоть до 4х десятичных знаков). Пример: `calc("девятнадцать и восемьдесят две сотых разделить на девяносто девять")` -> "ноль и двадцать сотых и ноль два в периоде".

#### **Сложность 3**

- 3) Реализовать текстовый калькулятор для выражения из произвольного количества операций с учетом приоритета операций. Пример: `calc("пять плюс два умножить на три минус один")` -> "десять". (Для реализации рекомендуется использовать алгоритмы основанные на польской инверсной записи см. например, [https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F\\_%D0%BF%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%B0%D1%8F\\_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C) )

**Сложность 3**

- 4) *Расширение задания 3.* Добавить поддержку приоритета операций с помощью скобок. Пример: `calc("скобка открывается пять плюс два скобка закрывается умножить на три минус один")` -> "двадцать".

**Сложность 3**

- 5) Добавить возможность использования отрицательных чисел. Пример: `calc("пять минус минус один")` -> "шесть".

**Сложность 1**

- 6) Добавить возможность оперировать с дробями (правильными и смешанными). Реализовать поддержку сложения, вычитания и умножения, дробей. Результат операций не должен представлять неправильную дробь, такие результаты нужно превращать в смешанные дроби. Пример: `calc("один и четыре пятых плюс шесть седьмых ")` -> "два и двадцать три тридцать пятых".

**Сложность 3**

- 7) *Расширение задания 6.* Добавить автоматическое сокращение дроби в ответе. Пример: `calc("одна шестая умножить на две третьих")` -> "одна девятая".

**Сложность 1**

- 8) *Расширение задания 1.* Добавить операции возведения в степень и тригонометрические операции синус, косинус, тангенс и константу пи. Допускается как минимум одна из этих функций в выражении с обычными операциями. Пример: `calc("два в степени четыре")` -> "шестнадцать". Пример: `calc("синус от пи разделить на четыре")` -> "ноли и семьсот семь тысячных".

**Сложность 1 или 2**

- 9) Добавить комбинаторные операции перестановки, размещения и сочетания. Пример: `calc("размещений из трех по два")` -> "шесть".

**Сложность 1 или 2**

- 10) Диагностировать ошибки: неправильную запись числа; неправильную последовательность чисел и операций; (задание 1) деление на ноль; (задание 3) неправильную последовательность чисел и операций; (задание 4) некорректный баланс и вложенность скобок; (задание 6) некорректную запись числа

**Сложность 1 или 2**

Для каждого студента формируется комплексное задание из сочетания пунктов. Суммарная сложность комплексного задания должна быть не менее 7. Все выбранные пункты должны работать в функции `calc()` одновременно.

Приветствуется выполнение заданий с суммарной сложностью более 7 (рекомендуется отмечать такие решения дополнительными баллами). Если реализовано 2 и более пунктов сложности 3, то, считается, что выполнено задание со звездочкой.

Сложность пункта с интервальной сложностью (например: Сложность 1 или 2) определяется преподавателем в зависимости от сочетания этого пункта с другими пунктами, выбранными студентом. В случае неоднозначности сочетания требований двух пунктов студент предлагает преподавателю свое видение итоговой постановки задачи, а преподаватель согласует его (рекомендуется фиксировать договоренность в письменном виде). Преподаватель имеет право увеличить оценку сложности комплексного задания в случае существенного увеличения трудоемкости при сочетании двух пунктов.

Преподаватель имеет право добавить пункты с собственной постановкой задачи и оценкой сложности. Студент может предложить свои пункты и реализовывать их в случае предварительно согласования с преподавателем постановки задачи и оценки сложности пункта. Рекомендуется

**фиксировать дополнительные пункты письменно. Желательно, чтобы дополнительные пункты составляли не более половины суммарной трудоемкости комплексного задания.**