Day 00. Процедурный C#

# Общие требования

* Убедитесь, что на вашем компьютере установлен [SDK для разработки на .NET 5](https://dotnet.microsoft.com/download) и вы используете именно его.
* Помните, ваш код будут читать! Обратите особое внимание на оформление вашего кода и именование переменных. Придерживайтесь общепринятых стандартов [C# Coding Conventions](https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/inside-a-program/coding-conventions).
* Самостоятельно выберите удобную для себя IDE.
* Программа должна иметь возможность запуска через командную строку dotnet.
* В каждом из заданий указаны входные параметры и формат ответа на выходе. Необходимо придерживаться их.
* В начале каждого задания приведен список разрешенных языковых конструкций.
* Если затрудняетесь в решении задачи, обратитесь с вопросами к другим участникам бассейна, интернету, Google, посмотрите на StackOverflow.
* Избегайте хардкода и “магических чисел”.
* Вы демонстрируете все решение, верный результат работы программы — лишь один из способов проверки ее корректной работы. Поэтому когда необходимо получить определенный вывод в результате работы ваших программ, запрещено показывать предрассчитанный результат.
* Have fun :)

# Требования к заданиям дня

* Все сборки должны быть в одном решении.
* Каждому из заданий должно соответствовать отдельное консольное приложение, созданное на основе стандартного шаблона .NET SDK.

Так как в первом занятии мы рассматриваем язык как процедурный:

* Нельзя использовать классы.
* Нельзя использовать nuget-пакеты.

# Задание 00. Кредитный калькулятор

## Разрешенные языковые конструкции

* Примитивные типы
* Массивы
* Top-level-statements
* Локальные функции
* System.Math
* System.DateTime
* Интерполяция строк
* Циклы
* Parse/TryParse

## Структура проекта

Program.cs

## Задание

Представьте, что для обучения в Школе 21 пришлось временно оставить работу и, чтобы не остаться без средств к существованию, вы решили взять кредит. Так как вы любите все просчитывать наперед, вам важно узнать, какой из способов погашения кредита позволит расплатиться с банком с меньшей переплатой: уменьшение суммы платежа или сокращение срока кредита.

В кредитном договоре описаны формулы, по которым банк производит вычисления:

|  |
| --- |
| Ежемесячный платеж:  n — количество месяцев, в которые выплачивается кредит.  i — процентная ставка по займу в месяц.  Процентная ставка по займу в месяц:  Формула для расчета процентов ежемесячного платежа:  Сумма общего долга - сумма основного долга на дату расчета.  Число дней периода - разность в днях между датами «Дата текущего платежа» и дата предыдущего платежа.  Количество месяцев кредита: |

Итак, у вас есть кредит с определенной процентной ставкой и сроком. Дата первого платежа — 1 число следующего месяца. Что выгоднее: уменьшить срок кредита или сумму платежа?

### Досрочное погашение с уменьшением суммы

Ниже приведена таблицы контрольных значений - графики платежей для кредита на сумму 1 000 000 рублей на 10 месяцев со ставкой 12%. Кредит взят в мае, и дата первого платежа - 1 июня. При подсчете переплаты считаем, что вносим досрочный платеж 100 000 рублей на 5 месяце выплат. Т.е срок останется тем же, а ежемесячный платеж уменьшится.

Ежемесячный платеж состоит из суммы, которая каждый месяц идет в счет погашения общего долга (ОД), и части аннуитетного платежа, которая уходит в счет оплаты процентов по кредиту каждый месяц (в табл. Проценты). Проценты начисляются на остаток общего долга.

Таким образом, зная формулу расчет ежемесячного платежа, сумму, ОД можно найти как разницу между процентами и суммой ежемесячного платежа.

Для *досрочного погашения с уменьшением суммы* необходимо уменьшить Сумму общего долга (Остаток долга) на дату расчета и пересчитать сумму аннуитетного платежа (Платеж) и процентов для оставшихся месяцев кредита. Таким образом, переплата будет разницей между той суммой, которую банк дал вам в кредит, и суммой платежей за все месяцы кредита.

|  |
| --- |
| Дата Платеж ОД Проценты Остаток долга  105 582,08р. 1 000 000,00 р. 55 820,77 р. 1 000 000,00 р.  01.06.2021 105 582,08 р. 95 390,30 р. 10 191,78 р. 904 609,70 р.  01.07.2021 105 582,08 р. 96 659,90 р. 8 922,18 р. 807 949,81 р.  01.08.2021 105 582,08 р. 97 347,63 р. 8 234,45 р. 710 602,18 р.  01.09.2021 105 582,08 р. 98 339,77 р. 7 242,30 р. 612 262,40 р.  01.10.2021 85 036,56 р. 99 543,32 р. 6 038,75 р. 412 719,08 р.  01.11.2021 85 036,56 р. 80 830,21 р. 4 206,34 р. 331 888,86 р.  01.12.2021 85 036,56 р. 81 763,13 р. 3 273,42 р. 250 125,73 р.  01.01.2022 85 036,56 р. 82 487,33 р. 2 549,23 р. 167 638,40 р.  01.02.2022 85 036,56 р. 83 328,02 р. 1 708,53 р. 84 310,38 р.  01.03.2022 85 036,56 р. 84 260,44 р. 776,12 р. 49,94 р. |

В этом случае *при уменьшении суммы* переплата: 53 143,10р.

### Досрочное погашение с уменьшением срока

Ниже приведена таблица контрольных значений - график платежей для кредита на сумму 1 000 000 рублей на 10 месяцев со ставкой 12%. При подсчете переплаты считаем, что вносим досрочный платеж 100 000 рублей на 5 месяце.

Для досрочного погашения с уменьшением срока, необходимо уменьшить Сумму общего долга на дату расчета и вычислить новое количество месяцев, которое осталось до полной выплаты кредита.

Для расчета числового значения переплаты необходимо найти разницу между той суммой, которую банк дал вам в кредит, и суммой платежей за общее количество месяцев кредита с учетом пересчета срока.

Допустим, вы планируете внести досрочно 100 000 на 5-й месяц кредита.

|  |
| --- |
| Дата Платеж ОД Проценты Остаток долга  105 582,08р. 1 000 000,00 р. 55 820,77 р. 1 000 000,00 р.  01.06.2021 105 582,08 р. 95 390,30 р. 10 191,78 р. 904 609,70 р.  01.07.2021 105 582,08 р. 96 659,90 р. 8 922,18 р. 807 949,81 р.  01.08.2021 105 582,08 р. 97 347,63 р. 8 234,45 р. 710 602,18 р.  01.09.2021 105 582,08 р. 98 339,77 р. 7 242,30 р. 612 262,40 р.  01.10.2021 105 582,08 р. 99 543,32 р. 6 038,75 р. 412 719,08 р.  01.11.2021 105 582,08 р. 101 375,73 р. 4 206,34 р. 311 343,34 р.  01.12.2021 105 582,08 р. 102 511,29 р. 3 070,78 р. 208 832,05 р.  01.01.2022 105 582,08 р. 103 453,71 р. 2 128,37 р. 105 378,34 р.  01.02.2022 105 582,08 р. 104 508,08 р. 1 073.99 р. 870,26 р.  01.03.2022 105 582,08 р. 105 574,07 р. 8,01 р. 0,00 р. |

Переплата: 51 108,95р.

Итак, вам нужно написать консольное приложение-калькулятор, которое поможет вам принять решение.

### Входные параметры

|  |  |  |
| --- | --- | --- |
| sum | double | Сумма кредита, р |
| rate | double | Годовая процентная ставка, % |
| term | int | Количество месяцев кредита |
| selectedMonth | int | Номер месяца кредита, в котором вносится досрочный платёж |
| payment | double | Сумма досрочного платежа, р |

### Формат ответа на выходе

#### Уменьшение срока выгоднее уменьшения платежа

Переплата при уменьшении платежа: {сумма переплаты}р.

Переплата при уменьшении срока: {сумма переплаты}р.

Уменьшение срока выгоднее уменьшения платежа на {разница}р.

#### Уменьшение платежа выгоднее уменьшения срока

Переплата при уменьшении платежа: {сумма переплаты}р.

Переплата при уменьшении срока: {сумма переплаты}р.

Уменьшение платежа выгоднее уменьшения срока на {разница переплаты}р.

#### Уменьшение платежа равно уменьшению срока

Переплата при уменьшении платежа: {сумма переплаты}р.

Переплата при уменьшении срока: {сумма переплаты}р.

Переплата одинакова в обоих вариантах.

#### Пользователь указал некорректные данные

Ошибка ввода. Проверьте входные данные и повторите запрос.

#### Пример запуска приложения из папки проекта и вывода

$ dotnet run 1000000 12 10 5 100000

Переплата при уменьшении платежа: 53 143,10р.

Переплата при уменьшении срока: 51 108,95р.

Уменьшение срока выгоднее уменьшения платежа на 2 034,15р.

### Бонусное задание

1. Не менее удобным был бы расчет и вывод на экран графика платежей для обоих случаев. Примеры приведены в описании. Таблица должна иметь столбцы: Номер месяца, Дата платежа, Общая сумма платежа (Платеж), Тело кредита (ОД), Проценты, Остаток долга.

# Задание 01. Исправление опечаток

## Разрешенные языковые конструкции

* Примитивные типы
* Массивы
* Top-level-statements
* Локальные функции
* Циклы
* System.IO
* System.Math
* Regex
* Интерполяция строк

## Структура проекта

Program.cs

## Задание

А теперь представьте, что вы нашли неплохую работу с графиком, который позволит вам комфортно обучаться в Школе 21 и оплачивать кредит, да еще и оттачивать полученные в школе навыки. И вот на новом месте вам приходит первая задача:

“Реализовать автокоррекцию имени пользователя”

Дело в том, что пользователи системы не всегда внимательно относятся к заполнению форм и допускают опечатки. Но есть словарь, содержащий список всех имен, по которому можно проверить верность написания, и в ответ на ошибку в имени пользователя можно предложить исправление. Это вам и нужно реализовать.

У вас есть файл со [списком всех англоязычных имен](https://github.com/qdimka/NameDatabases/blob/master/NamesDatabases/first%20names/us.txt), из которого нужно считать список имен для проверки. Здесь мы считаем, что словарь полный и содержит все возможные варианты (если пользователь ничего не выбирает или имя не найдено, выводится ошибка). Для сравнения слов вы решаете использовать [расстояние Левенштейна](https://en.wikipedia.org/wiki/Levenshtein_distance). Считаем имя близким к введенному, если расстояние редактирования до него менее 3.

Итак, вы должны реализовать консольное приложение, которое запрашивает имя пользователя:

>Enter name:

Далее пользователь должен ввести свое имя, на что программа реагирует следующим образом:

|  |  |
| --- | --- |
| Имя найдено в словаре | Hello, {имя}! |
| В словаре найдено близкое имя (расстояние редактирования до которого не более 2) | >Did you mean “{исправленное имя}”? Y/N |
| Y | Hello, {исправленное имя}! |
| N | * >Did you mean “{новый вариант исправления}”? Y/N * Your name was not found. |
| Близкое имя не найдено | Your name was not found. |
| Имя не введено | Your name was not found. |

А еще было бы здорово заранее проверять имя пользователя на верный ввод. Ведь оно может содержать только буквы, пробелы и дефисы.

Пример запуска приложения из папки проекта и вывода

$ dotnet run

>Enter name:

Mrk

>Did you mean “Mark”? Y/N

Y

>Hello, Mark!