|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Начальные баланс, р | Зарплата, р | Сумма долга, р | Процентная ставка, % |
| Вася | 2750 | 100 | 3900 | 6.3 |
| Петя | 2500 | 170 | 3800 | 5.6 |
| Коля | 1800 | 210 | 3200 | 6.8 |
| Таня | 2200 | 150 | 3750 | 4.9 |
| Маша | 2000 | 190 | 4000 | 6.4 |

Необходимо рассчитать, за сколько месяцев должники смогут погасить свои долги банку. Люди имеют зарплату, которая начисляется каждый месяц. После этого они платят все имеющиеся деньги в банк. Если долг превышает лимит банковской лояльности 1000 р, то долг увеличивается на процентную ставку.

Задания:

1. Создать структуру и занести в неё данные из таблицы (кроме серого столбца).
2. Рассчитать, сколько месяцев понадобится каждому из людей для погашения долга. Вывести имя человека и количество месяцев на экран.
3. Поменять структуру на класс и добавить класс-наследник.
4. Сделать увеличение долга каждый месяц, если долг больше банковской лояльности (1000 р)
5. Сделать сортировку должников по возрастанию времени выплат. Если количество месяцев у людей совпадает, отсортировать их по остаточному балансу. Записать конечную таблицу в файл.

Разобьем задачу на подзадачи:

1. Составить структуру с конструктором по первым 4-м колонкам таблицы. Не забыть, что там же надо создать подсчет месяцев.
2. Создать методы начисления зарплаты и погашения долга.
3. В основной программе пройтись по всем людям и вызывать методы, пока долг не будет погашен.
4. Вывести на консоль всех людей, чтобы посмотреть, что у нас получилось. Интересуют только имена и количество месяцев.
5. Показать результат преподавателю.
6. Изменить структуру на класс.
7. Добавить наследника, конструктор наследника и обращение к базовому конструктору. В основной программе изменить тип передаваемых данных и добавить данные из 5-го столбца.
8. Сделать у наследника метод начисления процентов к долгу. Добавить начисление процентов в основной программе после акта погашения.
9. Показать результат преподавателю.
10. Сделать метод сортировки в основной программе. Учесть, что, если условие сортировки не выполняется, следует проверить другое условие.
11. Добавить библиотеку System.IO и открыть поток для записи в файл. В качестве пути можно использовать универсальный доступ к рабочему столу, сетевой диск, диск С.
12. Запустить программу, проверить, совпадают ли данные в записанном файле с ожидаемыми.
13. Показать результат преподавателю.

Далее будет 2 примера для самостоятельного решения

Необходимо рассчитать наибольшую выгоду. Принять, что игрушки имеют одинаковый размер. Вместимость склада 200 игрушек. Бюджет составляет 30 000 р. Каждый поставщик продает игрушки партиями. За покупку всех 100 шт., он предоставляет скидку. Больше 100 шт. каждого товара купить невозможно. Каждый месяц меняется популярный товар (случайно), тогда его можно продать подороже.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Стоимость р/шт | Количество в партии, шт | Скидка за опт (100 шт), % | Цена | Бонус, % |
| Самосвалы | 220 | 10 | 10 | 330 | 5 |
| Поезда Ту-ту | 200 | 20 | 5 | 240 | 10 |
| Плюшевые мишки | 250 | 25 | 15 | 310 | 8 |

Задания:

1. Создать структуру и занести в неё данные из таблицы (кроме серого столбца).
2. Рассчитать, сколько нужно купить игрушек каждого вида, чтобы получить максимальную прибыль. Вывести названия и количества игрушек в виде таблицы. А ниже максимальную выгоду, которую можно получить.
3. Поменять структуру на класс и добавить класс-наследник (серый столбец).
4. Сделать, чтобы при каждом запуске выбирался случайный товар, на который распространяется бонус популярности. Совершить расчет игрушек с учетом этого бонуса.
5. Отсортировать игрушки по приобретенному их количеству. Записать конечную таблицу в файл.

Необходимо рассчитать оптимальную программу тренировок для студентов с разным уровнем подготовки. Общая продолжительность тренировки должна не превышать 60 минут. Тренировка рассчитана на 7 дней (одна и та же тренировка каждый день). Студенты должны потратить все лишние ккал, что у них есть, но не уйти в минус более, чем на 1000 ккал. Студент не может выполнять тренировку, которая превышает уровень его подготовки (для заданий 3-5). Студент не может выполнять одно и тоже упражнение более, чем 2 раза.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Количество сжигаемых в минуту, ккал | Время тренировки, мин | Степень сложности |
| Бег | 24 | 15 | 5 |
| Плавание | 17 | 25 | 6 |
| Велоспорт | 13 | 60 | 8 |

|  |  |  |
| --- | --- | --- |
| Студент | Количество лишних ккал | Уровень подготовки |
| Даня | 5000 | 12 |
| Коля | 5900 | 12 |
| Саша | 8000 | 16 |
| Стас | 5500 | 15 |
| Даша | 3000 | 8 |

Задания:

1. Создать 2 структуры и занести в них данные из таблицы (кроме серых столбцов).
2. Рассчитать, сколько ккал сжигается за тренировку каждого типа. Вывести названия упражнений в тренировку для каждого студента.
3. Поменять структуры на классы и добавить им классы-наследники (серые столбцы).
4. Рассчитать, сколько ккал сжигается за тренировку каждого типа с учетом ограничений по уровню подготовки.
5. Записать конечную таблицу в файл.