

Задача (Task) 4. Варианты задачи

1. Термометр

Разработать класс **Thermometer**.

Класс должен хранить историю наблюдений за температурой в течение одного календарного (не високосного) года. Наблюдения описываются датой (день, месяц, год) и временем (в часах). При поступлении нового наблюдения для уже существующих даты и времени старое наблюдение заменяется. Температура задается в градусах Цельсия.

Начальная дата и время наблюдений устанавливается при конструировании объектов и в процессе работы не меняется.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор, ...).

Класс должен предоставлять следующие операции:

8) сохранить историю наблюдений в файл и считать историю наблюдений из файла.

1) узнать начальные дату и время наблюдений,

2) добавить наблюдение,

3) узнать температуру в наблюдении, выбранном по дате и времени,

4) выдать серию наблюдений для выбранной даты (не на консоль),

5) найти среднюю температуру для выбранной даты, выбранного месяца, за всю историю наблюдений,

7) найти среднюю дневную или ночную температуру для выбранного месяца,

Программа должна иметь простое меню для выбора необходимых операций.

2. Весы напольные

Разработать класс **FloorScales**.

Класс должен хранить историю наблюдений за показаниями веса членов семьи (до пяти человек). Показания описываются датой (день, месяц, год) и именем члена семьи. При поступлении нового наблюдения для уже существующей даты старое наблюдение заменяется. Вес задается в килограммах с точностью до 50 граммов.

Начальная дата и время наблюдений устанавливается при конструировании объектов и в процессе работы не меняется.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор, ...).

Класс должен предоставлять следующие операции:

7) сохранить историю наблюдений в файл и считать историю наблюдений из файла.

1) узнать начальные дату и время наблюдений,

2) задать наблюдение,

3) узнать вес в наблюдении, выбранном по дате и имени члена семьи,

4) найти средний вес члена семьи в выбранном месяце или за всю историю наблюдений,

5) найти минимальный вес члена семьи в выбранном месяце или за всю историю наблюдений и дату, когда он наблюдался

6) найти максимальный вес члена семьи в выбранном месяце или за всю историю наблюдений и дату, когда он наблюдался,

Программа должна иметь простое меню для выбора необходимых операций.

3. Шагомер

Разработать класс **Pedometer**.

Класс должен хранить историю подсчета шагов владельца. Каждый подсчет описывается датой (день, месяц, год) и интервалом времени (час, минута начала движения, час, минута окончания движения). Подсчет ведется в шагах с точностью до единицы.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор, ...).

Начальная дата и время наблюдений устанавливается при конструировании объектов и в процессе работы не меняется.

Класс должен предоставлять следующие операции:

6) сохранить историю подсчетов в файл и считать историю подсчетов из файла.

1) узнать начальные дату и время подсчетов,

2) добавить подсчет,

3) получить информацию о подсчете, выбранном по дате и интервалу времени,

4) найти среднее число шагов в выбранном месяце или за всю историю наблюдений,

5) найти максимальное число шагов в день в выбранном месяце или за всю историю наблюдений и дату, когда оно было достигнуто,

Программа должна иметь простое меню для выбора необходимых операций.

4. Контакты

Разработать класс **Contacts**.

Класс должен хранить информацию о контактах владельца. Каждый контакт содержит следующие данные: фамилия; имя; отчество; телефон; день рождения (день, месяц, год); признак, относится ли контакт к избранным. Контакты хранятся упорядоченно по фамилии, имени, отчеству.

Фамилия, имя, отчество (ФИО) являются обязательными полями. Данные вводятся на русском языке.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор, ...).

Класс должен предоставлять следующие операции: ,

11) сохранить контакты в файл и считать контакты из файла.

1) создать новый контакт, 2) найти контакт по ФИО; 3) изменить контакт, выбранный по ФИО, 4)

найти контакт по телефону, 5) выдать все контакты на заданную начальную букву ФИО (не на консоль), 6) узнать текущее число контактов, 7) внести контакт, выбранный по ФИО в список избранных, 8) удалить контакт, выбранный по ФИО из списка избранных, 9) выдать все избранные контакты (не на консоль), 10) удалить контакт, выбранный по телефону

Программа должна иметь простое меню для выбора необходимых операций.

5. Песенник

Разработать класс **Songs**.

Класс должен хранить информацию о песенных композициях. Каждая песня описывается следующими данными: название, поэт (автор стихов), композитор (автор музыки), исполнитель, название альбома (если входит в какой-то альбом), дата выпуска (день, месяц, год). Песни хранятся упорядоченно по названию. Данные вводятся на русском языке.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор, ...).

Класс должен предоставлять следующие операции:

9) сохранить песенник в файл и считать песенник из файла.

1) добавить песню, 2) изменить данные песни, выбранной по названию, 3) найти песню по

названию и исполнителю, 4) выдать все песни заданного поэта (не на консоль), 5) выдать все

песни заданного композитора (не на консоль), 6) выдать все песни заданного исполнителя (не на консоль), 7) узнать текущее число песен, 8) удалить песню по названию,

Программа должна иметь простое меню для выбора необходимых операций.

6. Фильмотека

Разработать класс **FilmLibrary**.

Класс должен хранить информацию о фильмах. Каждый фильм описывается следующими данными: название, режиссер, сценарист, композитор, дата выхода в прокат (день, месяц, год),

сборы (в рублях). Фильмы хранятся упорядоченно по названию и годам. Данные вводятся на русском языке.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор, ...).

Класс должен предоставлять следующие операции:

10) сохранить фильмотеку в файл и считать фильмотеку из файла.

1) добавить фильм, 2) изменить данные фильма выбранного по названию, 3) найти фильм по названию и году, 4) выдать все фильмы заданного режиссера (не на консоль),, 5) выдать все фильмы, вышедшие в прокат в выбранном году (не на консоль),, 6) выдать заданное число фильмов с наибольшими сборами (не на консоль),, 7) выдать заданное число фильмов с наибольшими сборами в выбранном году (не на консоль),, 8) узнать текущее число фильмов, 9) удалить фильм по названию,

Программа должна иметь простое меню для выбора необходимых операций.