**Цель:**

Создать программу, которая сделает вывод, похудеет человек от количества съеденного или нет.

**Условия**

Для работы с программой использовать введение информации с помощью System.in (Scanner). Пример работы с Scanner представлен в конце документа.

Классы в программе должны были разделены логически по папкам.

При выборе неправильного индекса, типов атрибутов, каких-то других параметров, должно срабатывать исключение с пояснением ошибки.

Функционал может и, возможно, должен быть расширен, но обязан в себе содержать:

Класс продукт:

* свойства класса: Имя продукта, Белки, Углеводы, Жиры;
* конструктор;
* get методы на них;
* метод, который будет считать калории. Данный метод должен на вход получать n количество грамм и возвращать m количество калорий.

Класс человек:

* Атрибуты класса: имя, рост, вес, возраст, пол, коэффициент активности, количество потребляемых калорий.
* Конструктор;
* Get и set методы на них;
* Коэффициент активности должен устанавливаться через set функцию, которая на вход получает номер коэффициента активности. Номер коэффициента должен быть не меньше 0 и не больше 6;
* Метод расчета DCI;
* Метод, который делает вывод, худеет человек или нет;
* Метод «новый день», позволяющий установить количество потребляемых калорий на 0.

Меню программы:

С помощью Scanner’a сделать выбор между возможностями программы

* 1. Возможность установить с помощью Scanner’a добавление или выбор людей для просмотра их сравнения DCI с установленным рационом. Для удобства выбора пользователю в консоли должны быть перечислены уже созданные люди и их индексы.
  2. Возможность установить с помощью Scanner’a отчистку и добавление съеденных калорий для любого из созданных людей. Для удобства пользователю в консоли должны быть перечислены внесённые в программу продукты.
  3. Выход из программы. Т.е. программа должна работать до тех пор, пока не будет выбран пункт выхода из программы.

Загрузка продуктов должна быть осуществлена через файл. Если файл не найден, должна быть выведена соответствующая ошибка. В данном случае программа должна работать с заложенной в ней 5 продуктами.

В программе должны быть созданные 4 человека с уже потребленным набором блюд.

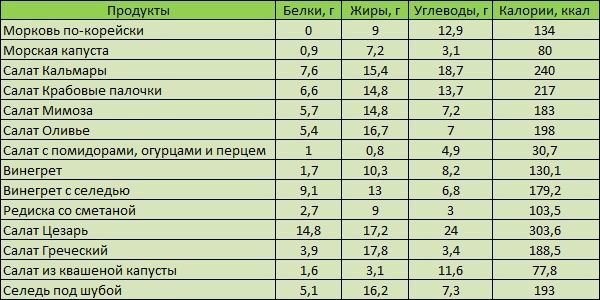
**Материалы для программы**

Пищевая ценность

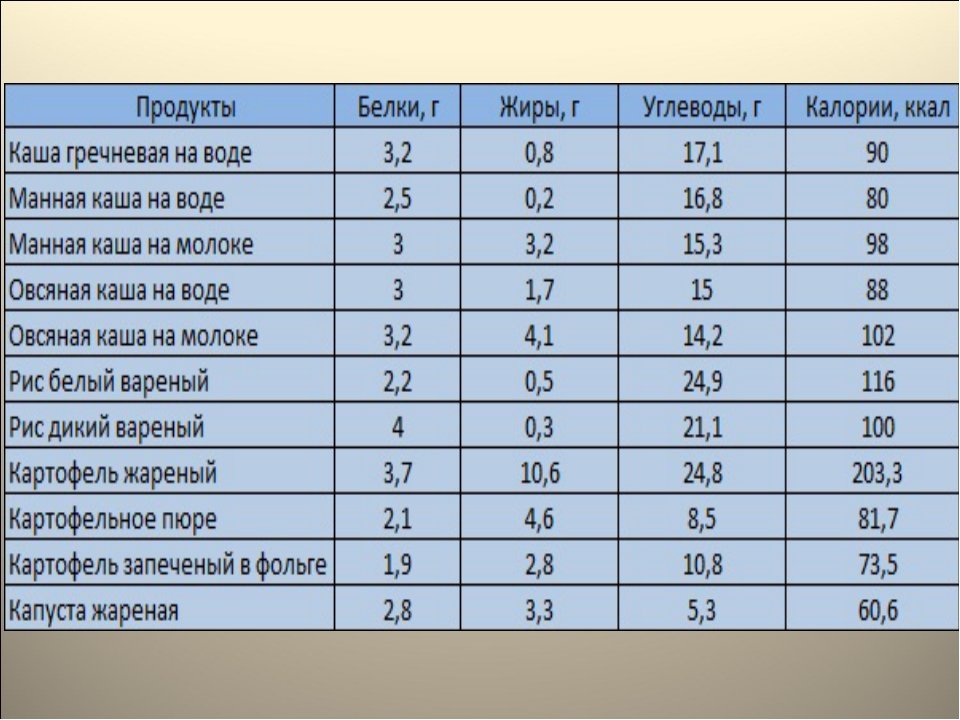
|  |  |
| --- | --- |
| Компонент пищи | ккал/г |
| Жиры | 9,29 |
| Белки | 4,1 |
| Углеводы | 4,1 |



Список продуктов









Пример использования Scanner

Для работы с Scanner, необходимо установить пакет java.util.Scanner

Для создания самого объекта Scanner в его конструктор передается объект System.in. После этого мы можем получать вводимые значения. Например, в данном случае вначале выводим приглашение к вводу и затем получаем вводимое число в переменную num.

Чтобы получить введенное число, используется метод **in.nextInt();**, который возвращает введенное с клавиатуры целочисленное значение.

Класс Scanner имеет еще ряд методов, которые позволяют получить введенные пользователем значения:

* **next()**: считывает введенную строку до первого пробела
* **nextLine()**: считывает всю введенную строку
* **nextInt()**: считывает введенное число int
* **nextDouble()**: считывает введенное число double
* **nextBoolean()**: считывает значение boolean
* **nextByte()**: считывает введенное число byte
* **nextFloat()**: считывает введенное число float
* **nextShort()**: считывает введенное число short

Пример работы для создания человека

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Input name: ");

String name = in.nextLine();

System.out.print("Input age: ");

int age = in.nextInt();

System.out.print("Input height: ");

float height = in.nextFloat();

System.out.printf("Name: %s Age: %d Height: %.2f \n", name, age, height);

in.close();

}

}