

Task1

На странице представлена информация по заданию Task1 (лабораторная работа №1).

- [Входные данные](#)
 - [Проект заглушка](#)
 - [Корневой пакет проекта](#)
- [Задание](#)
 - [Формулировка](#)
 - [Замечания](#)
- [Порядок выполнения работы](#)
- [Вопросы \(допуск\)](#)

Входные данные


Проект заглушка

Проект заглушка Task1Stub находится в репозитории по адресу: /examples/projects/Task1Stub


Корневой пакет проекта

Корневой пакет для проекта: ua.nure.**yourlastname**.Task1

Все классы должны располагаться в этом пакете.

 **yourlastname** заменить на свою фамилию (ваш логин, без последних трех букв, которые обозначают код проекта - jff).

Задание

 Во всех задачах входная информация поступает в виде параметров командной строки. Выходная информация, которую генерирует задача, поступает в стандартный поток вывода (консоль).

Формулировка

1. Написать программу, которая находит наибольший общий делитель двух целых положительных чисел.

Название класса: Part1

Вход: X Y

Выход: наибольший общий делитель X и Y

2. Написать программу, которая находит сумму цифр заданного целого числа.

Название класса: Part2
Вход: X
Выход: сумма цифр числа X

3. Написать программу проверки того, что заданное число простое (т.е. не делится без остатка ни на какие числа, кроме себя и 1).

Название класса: Part3
Вход: X
Выход: true - если X простое число; false - если X не является простым числом

4. Подсчитать сумму ряда $1! - 2! + 3! - 4! + 5! - \dots \pm N!$ для заданного числа $N > 0$.

Название класса: Part4
Вход: N
Выход: сумма ряда $1! - 2! + 3! - 4! + 5! - \dots \pm N!$

5. Подсчитать, сколько N-значных чисел (N-четное, больше 1) имеют равную сумму первой половины и второй половины цифр ("счастливые" числа). Пример: вход N=6, выход - количество шестизначных чисел ABCDEF (ведущий ноль не допускается), у которых $(A+B+C) = (D+E+F)$

Название класса: Part5
Вход: N
Выход: количество N-значных "счастливых" чисел

6. Разместить в памяти массив из N элементов и заполнить его рядом Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, В этом ряду каждое следующее число является суммой двух предыдущих.

Название класса: Part6
Вход: N
Выход: N первых элементов ряда Фибоначчи, разделенные пробелом

Пример
Вход: 7
Выход: 1 1 2 3 5 8 13

7. Создать целый массив из N элементов и заполнить его простыми числами: 2, 3, 5, 7, 11, 13, 17.

Название класса: Part7
Вход: N
Выход: первые N простых чисел в ряде натуральных чисел, разделенные пробелами

Пример
Вход: 5
Выход: 2 3 5 7 11

8. Создать двумерный массив ($N \times M$) символов и заполнить его буквами 'Ч' и 'Б' в шахматном порядке.

Название класса: Part8

Вход: N M

Выход: таблица из чередующихся кириллических букв 'Ч' и 'Б' в шахматном порядке

Пример

Вход: 3 5

Выход:

ЧБЧБЧ

БЧБЧБ

ЧБЧБЧ

9. Создать целый пятимерный массив с двумя значениями в каждом измерении. Заполнить массив числами из начала натурального ряда: 1, 2, 3, При заполнении использовать один цикл.

Название класса: Part9

Вход: отсутствует

Выход: 1 2 3 ... 32

10. Создать "треугольный" массив из N строк и заполнить его биномиальными коэффициентами.

Название класса: Part10

Вход: N

Выход: биномиальные коэффициенты разделенные пробелами

Пример

Вход: 5

Выход:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1


Замечания

1. Проект в IDE (Eclipse/IDEA) должен иметь название Task1.
2. Решение каждой из 10 задач оформить в виде отдельного класса с названиями - Part1, Part2, ..., Part10. Значение входных данных получать из командной строки. В случае, если входные данные заданы не были - выводить информационное сообщение об этом.
3. Дополнительно создать класс Demo, который демонстрирует работу всех 10 подпрограмм с некоторыми заданными значениями (задать самостоятельно).
4. Дополнительно написать командный файл (bat-файл в Windows или shell-скрипт в Linux), который компилирует и запускает на выполнение класс Demo. Скрипт сохранить в каталоге Task1/src. Вручную запустить скрипт, проверить его работоспособность.
5. Соблюдать code convention.

Порядок выполнения работы


1. Используя заглушку (заглушку использовать не обязательно, можно писать все "с нуля"):
 - а. Сделать checkout проекта заглушки из репозитория и отвязать его от узла [/examples/projects/Task1Stub](#)

- b. Переименовать корневой пакет проекта: yourlastname == > ваш логин, без последних трех букв, которые обозначают код проекта - jff.
 - c. Переименовать проект Task1Stub ==> Task1.
2. Написать классы, которые реализуют решение задач 1-10. Написать класс Demo.
 3. Создать командный файл для вашей OS.
 4. Привязать проект к нужному узлу в репозитории и сделать коммит проекта в репозиторий.

 Коммит не пройдет, если среди файлов, помещаемых в репозиторий, будет хотя бы один class-файл, т.е. class-файлы не коммитить.

Также в обязательном порядке в репозиторий должны быть помещены метафайлы IDE (для Eclipse - .project; для IDEA - Task1.iml), иначе Jenkins проект не соберет.

5. Добиться сборки проекта в Jenkins (после каждого коммита в репозиторий Jenkins пересобирает проект, если проект не собрался, то причину можно посмотреть в логах сборки).
6. Оптимизировать метрики проекта в Sonar (Blocker/Critical/Major issues должны быть по нулям, RCI как можно ближе к 100%).
7. Прийти на занятие и защитить свою работу.

 Если вы не успели пропустить проект через систему (Jenkins/Sonar) - сдавать лабораторную как есть, приносите решение, сдавайте, заливать в репозиторий и пропускать через систему позже. Необходимым (но не достаточным!) условием получения оценки более 60 по предмету является на момент получения зачета для всех л.р.:

1. присутствие в репозитории
2. успешная сборка в Jenkins
3. выведенные в ноль issues в Sonar

Вопросы (допуск)

1. Какие категории типов данных существуют в Java?
2. Перечислите примитивные типы данных.
3. Тип данных char, что хранит, область определения.
4. Напишите метод main (два варианта).
5. Укажите автоматические преобразования между примитивными типами
6. Укажите автоматические преобразования между примитивными типами.
7. При каких преобразованиях между примитивными типами возможна утрата информации?
8. Какие преобразования между типами вы знаете.
9. Что такое wrappers, autoboxing, autounboxing.
10. Напишите анонимный массив, массив константу, в чем отличие.
11. Напишите пример двумерного массива.

