



Титульный лист

Направление Естественные науки Инженерные науки
 Математика и информатика Социальные и гуманитарные науки
 Экономика и управление

Вариативный блок 1 2 3 4 5

Курс 1 2 3 4 5 отсутствует

Фамилия М Е Л Ь Н И К О В

Имя М И Х Е Й

Отчество Д Е Н И С О В И Ч

Дата рождения 1 3 0 1 2 0 0 2

Город участия Е К А Т Е Р И Н Б У Р Г

Аудитория Ф - 4 0 1

Телефон 8 9 8 2 3 0 9 6 4 3 0

Дата 0 5 0 2 2 0 2 4

Подпись

Пример
заполнения

А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф
Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я 1 2 3 4 5 6 7 8 9 0



ИЗУМРУД.СТУДЕНТ
ОЛИМПИАДА УРАЛЬСКОГО ФЕДЕРАЛЬНОГО УНИВЕРСИТЕТА



3101529100589

Проверочный лист

Заполняется участниками

Направление Естественные науки Инженерные науки
 Математика и информатика Социальные и
 Экономика и управление гуманитарные науки

Вариативный блок 1 2 3 4 5

Курс 1 2 3 4 5 отсутствует

Город участия **ЕКАТЕРИНБУРГ**

Заполняется организаторами

Количество доп. листов **1** Количество черновиков к проверке :
Время выхода с до :

Протокол проверки

Заполняется жюри

Номер задания	1	2	3	4	5	6	7	8	9	10
Балл члена жюри №1	4	3	7							
Балл члена жюри №2	4	3	7							

Итоговый балл **41**

Подпись
члена жюри №1

Пилатов

Подпись
члена жюри №2

[Signature]

Пример
заполнения

А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф
Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я 1 2 3 4 5 6 7 8 9 0



Задание 1 (про лису)

Лиса бесконечно много раз откусила сыр

Док-во:

1) т.к. лиса откусывала сыр через каждые $\frac{1}{n}$ минут $n \in [1; +\infty)$,
 то ~~сумма ряда~~ ~~с~~ $n \rightarrow \infty$ равна $2 \left(\sum_{n=1}^{\infty} \frac{1}{n} \right) = 2 \cdot \infty = \infty$ —

2) лиса каждый раз сначала уравнивала вес, а потом делала массу одного из кусков меньше.

если бы в какой-то момент лиса сделала бы массы ^{обоих} кусков > 0 , то она бы не смогла затем сделать массу одного из кусков меньше \Rightarrow противоречие

3) из 1) и 2) \Rightarrow лиса бесконечно много раз откусила сыр

Лисе досталось $3\frac{1}{2}$ кг сыра **ПОЧЕМУ? ЭТО НЕВЕРНОЕ утверждение.**

Док-во:

1) в самом начале лиса откусила у 1-го медвежонка 1 кг сыра

2) затем она ~~откусила~~ откусила $\sum_{n=1}^{\infty} b_n = \sum_{n=1}^{\infty} \frac{2}{n(n+2)}$, $n \in [1; +\infty)$ кг сыра
 сумма ряда: $\sum_{n=1}^{\infty} \frac{2}{n(n+2)} = 2 \sum_{n=1}^{\infty} \frac{1}{n(n+2)} \stackrel{HEI}{=} 2 \cdot \frac{1}{2} = 1$ кг —



Бланк ответов

3) получаем из а. 1) и 2) \Rightarrow Лиса в сумме откусила $3 \frac{1}{2} \text{ кг}$ сыра, т.к. откусила $\sum_{u=1}^5 b_u$ у каждого мишки

Медвежатам досталось поровну сыра = по $2 \frac{1}{2} \text{ кг}$ + 2 каждому

Реш.во:

1) из предыд. вопроса: сначала лиса откусила у 1-го 1 кг , а затем у каждого по $\sum_{u=1}^5 b_u = 1 \text{ кг}$

2) $M_1 = 4 \text{ кг} \Rightarrow 4 - 1 - \sum_{u=1}^5 b_u = 2 \text{ кг}$; $M_2 = 3 \text{ кг} \Rightarrow 3 - \sum_{u=1}^5 b_u = 2 \text{ кг}$

3) подробную таблицу $\binom{4}{1}$ см. в черновике стр 1

Лиса не может выбрать b_u так, чтобы первому досталось больше

Реш.во:

1) Лиса у каждого съедает за 2 мин. ровно b_u сыра

2) за первую минуту она съела у 1-го медведя 1 кг

3) значит должно быть: $M_1 = 4 - 1 - b_u > 3 - b_u = M_2$

$$\begin{array}{ccc} & \downarrow & + 2 \\ M_1 = 3 & > & 3 = M_2 \end{array} \text{ — неверно}$$

45



Задача 2 (про посудомойку)

3. Архитектура БД

posuda
"таблица имеющихся посуды"

posuda_type
"таблица с типами посуды"

id	идентифик.
type_id	внешн. ключ типа посуды
clean_level	уровень чистоты [0; 10], где 0 - чисто, 10 - грязно
deleted_at	дата-время удаления посуды (= void времени)

id	идентифик.
type	тип посуды = название
size_x	размер по X
size_y	размер по Y
size_z	размер по Z
deleted_at	/ / / / /
deleted_at	дата-время удаления типа

posuda_queue
"очередь посуды на мойку"

wash_machine
"описание полок машины"

id	идентификатор
posuda_id	внешн. ключ на посуду
priority	приоритет [1; 10], где 1 - малый, 10 - большой
cleared_at	решило дата-время работы отечистки
wash_machine_id	внешн. ключ на отсеки машины (= полка)
wash_queue_id	внешн. ключ на запуск машины

id	идентиф.
size_x	размер по X
size_y	размер по Y
size_z	размер по Z
power	мощность коэф

wash_queue
"очередь помывок (= запусков в машины)"

wash_option
"решилы старалки"

id	идентифик
wash_option_id	внешн. к. на решилы
start_at	начало дата-время
end_at	конец дата-время
posuda_queue_ids	список запуска посуды

id	идентифик
option	решилы (= название)
time	время работы
water	кол-во воды

~~posuda_wash_option_id
posuda_id
wash_option_id
wash_queue_id
wash_option_id~~

4. SELECT count(*) FROM wash_queue WHERE end_at IS NOT NULL;
-- кол-во моек

SELECT pt.type AS "тип", COUNT(*) FROM posuda_queue AS pq
LEFT JOIN posuda AS p ON pq.posuda_id = p.id
LEFT JOIN posuda_type AS pt ON p.type_id = pt.id
WHERE pq.cleared_at IS NOT NULL GROUP BY p.type_id; -- выводится
-- по типам



Алгоритм.

1) в основе лежит шагивый алгоритм, который учитывает:

~~1) уровень чистоты → берем сначала гряз~~

1) приоритет (posuda_grade.priority): берем сначала посуду с большим приоритетом

3) (уровень чистоты ~~→~~ мощность * режим) для каждой посуды
 $posuda.clean_level \leq wash_machine.power * wash_option.water$
 (считаем, что чем больше воды выльется на посуду, тем она чище)
 + хотим оптимизировать: $\sum_{\text{для всех машины}} water * power - clean_level \rightarrow 0$

2) подходит по размеру: $wash_machine.size_x, size_y, size_z \geq posuda.type.size_x, size_y, size_z$

+ хотим оптимизировать: $\sum_{\text{для всех машины}} (wm.size_x - pt.size_x) + \dots \rightarrow 0$
 [y, z]

2) шагивым алгоритмом с условиями 1), 2), 3) складываем

- посуду в машинку:
- 1) сначала по приоритету выбираем
 - 2) потом по размеру (→ max загроможденность)
 - 3) потом по чистоте (→ max чистоту и min воду) 20

Примеры:

- 1) если есть 3 тарелки с приоритетом 10 и 1 кружка с приоритетом 5, то сначала попробуем положить тарелки.
- 2) вылезает только 2 тарелки, но есть место для кружки, кладем ее тоже
- 3) чистота тарелки 10 (она грязнее), значит кладем ее на полку более мощную

2. Микрологическая модель (что это?)



- пользователь работает с системой через GUI
- GUI через API взаимодействует с БД
- есть очередь (Rabbit MQ), которая запускается раз в какое-то время или по триггеру и проверяет БД, а затем выполняет задачу по заданному алгоритму из п. 1

2

Плюсы решения:

- масштабируемость: можно менять данные о посуде, типах, машине
- можно проанализировать историю мойки конкретной посуды через две очереди
- можно написать на ~~PHP~~ Python и прикрутить умный дом