

Домашнее задание №3. Задача “Шулера”



«Шулера́». Микеланджело Меризи да Караваджо (около 1594 года)

Ваня заядлый игрок в карты. Ему очень нравится игра в “21”. Но так как ему не хватает аналитических способностей, он играет только в упрощенную версию игры со следующими правилами:

В игру всегда играют втроем. На столе бесконечная случайно перемешанная колода карт. Карты представляют собой целые числа в диапазоне от 1 до 10 (включительно). В начальный момент времени у игроков нулевой баланс. Игроки (в программе они *должны представлять собой потоки - threads*) пытаются взять карту из колоды **параллельно** и положить ее к себе в стопку, распикивая всех локтями. Взятие карты игроком приводит к увеличению баланса этого игрока на значение этой карты. После взятия карты игроки отдыхают (спят случайное время от 100 до 200 мс). В один момент времени **только один игрок** может взять карту, а остальные или отдыхают, или ждут, пока он её возьмет. Спустя 10 секунд после начала игры крупье **прерывает** игроков от попыток что-либо сделать и ждёт, когда все игроки успокоятся, а затем публикует баланс каждого игрока. После чего он (крупье) объявляет победителя (игрока с наибольшим балансом). В ходе игры игроки не знают баланс друг друга.

Ваня был удачлив и выиграл много денег, играя в эту игру, но однажды ему не повезло и он сел за стол с Колей и Петей..... А они оказались профессиональными **шулерами**.

Шулера (в отличие от честных игроков) вместо того, чтобы выполнить свой ход, могут воровать очки с баланса честных игроков (шулера всегда играют в команде) и записывать эти очки на свой баланс. Они выбирают такое действие с вероятностью 0.4 (40%) после отдыха. Но это действие намного сложнее, чем просто взять карту из колоды (и растолкать других игроков локтями), поэтому им требуется больше времени на отдых после этого (и они после этого спят случайное время от 180 до 300 мс). Шулера может своровать в каждой попытке от 0 до 8 очков, это количество каждый раз определяется случайно (если на балансе игрока не хватает очков, то крадется максимальное возможное количество; размер баланса игрока не должен быть отрицательным).

Надо понимать, что воровство баллов предполагает не только уменьшение баланса обворованного игрока, но и соответствующее увеличение баланса шулера.

Важно: на балансе лежат очки, а не карты, поэтому если Ваня вытащил две карты на 7 и на 3 очка, то его баланс равен 10, и шулер может украсть от 0 до 8 баллов (например, 5 баллов).

Для выполнения задачи можно использовать базовые средства синхронизации потоков, а именно — synchronized механизм.

Все остальные механизмы синхронизации (Lock, volatile и так далее) использовать **запрещено**.

Запуск программы:

```
java -jar main.jar
```

Функционал (логика в методах и классах) должен быть покрыт unit-тестами с использованием JUnit5. Покрывание тестами многопоточного кода не требуется.

Задание:

Оценка 6: реализована игра без шулеров (только три честных игрока).

Оценка 8: реализована игра с шулерами (двумя шулерами и одним честным игроком).

Оценка 10: реализована игра с переменным числом честных игроков и шулеров (в начале игры пользователь задает количество честных игроков (минимум один) и шулеров). Шулер случайным образом выбирает у кого воровать карты.

За отсутствие потоков - максимальная оценка 2.

При некорректной синхронизации потоков максимальная оценка 6.

Для проверки должен быть предоставлен архив с именем

<FamilyName_GivenName>_<groupName>_HW3.zip, в котором заархивирована

директория maven-проекта с pom.xml и всеми необходимыми для него каталогами, но без папки target (хранящей результаты компиляции и сборки). Распакованный проект должен собираться и тестироваться на машине проверяющего с установленным для проверки JDK 17 без дополнительных манипуляций (связанных с подключением библиотек, тестов и т.п.).

Например: Ivanov_Ivan_201_HW3.zip

Специально обращаем внимание на то, что в названии архива не должна использоваться кириллица (надо использовать только латиницу, как в примере)
Если архив назван не по гайдлайну: минус 2 балла от оценки.

Исходный код должен соответствовать java code convention:

<https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

За грубые нарушения допустимо снижение оценки (максимальное снижение 3 балла)