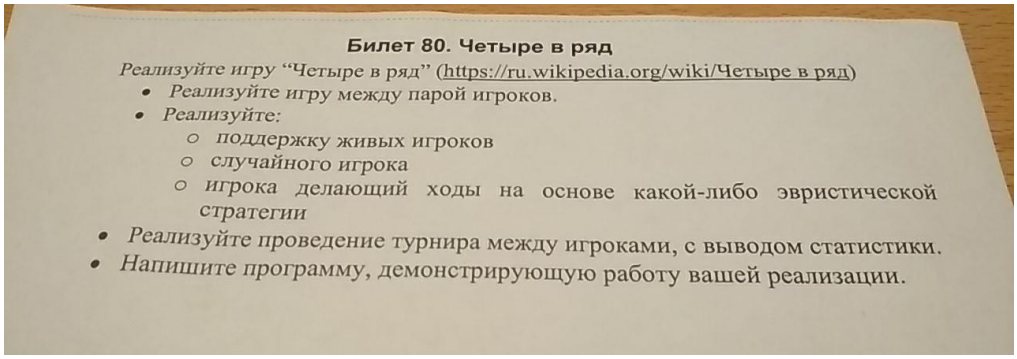
Билеты по программированию:

Билеты только на prolog идут только на одном языке, все остальные имеют вариации на трех языках (игры, интерпретаторы, парсеры, логгеры).

Мат ладей и королей	Ready(but with a lot of copy-paste)
КНФ	Ready
Реверси	Ready
Кубик Рубика	Not edit
Система Лейтнера	Not edit
Статистика по CSV	Not edit
Протоколирование	Not edit
Палочки	Ready
bbCodetoHTML	Ready
Spoon	Ready
Четыре в ряд	Ready
ДНФ	Ready
Автоматический переводчик	Ready
Проталкивание отрицаний	Ready
Grep	Ready(but with copy-paste)
Brainfuck	Ready
15	Ready
Линейная электронная таблица	Ready
Обобщенная строка	Ready
Игра Сет(prolog)	Ready
Судоку	None
Морской бой	None
Тракс(игра)	None
Шаблонизатор	None
Устойчивая сортировка	None
Жизнь	None
Калах	Not required
Шашки	Not required
Sort	Not required

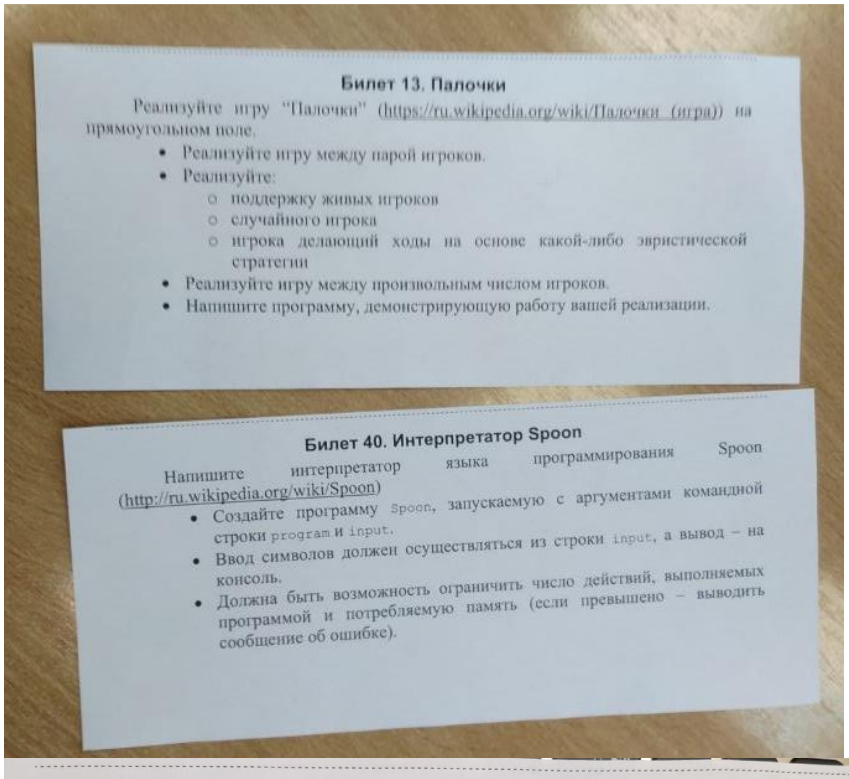
Фото билетов:



Билет 80. Четыре в ряд

Реализуйте игру "Четыре в ряд" (https://ru.wikipedia.org/wiki/Четыре в ряд)

- *Реализуйте игру между парой игроков.*
- *Реализуйте:*
  - *поддержку живых игроков*
  - *случайного игрока*
  - *игрока делающий ходы на основе какой-либо эвристической стратегии*
- *Реализуйте проведение турнира между игроками, с выводом статистики.*
- *Напишите программу, демонстрирующую работу вашей реализации.*



Билет 13. Палочки

Реализуйте игру "Палочки" (https://ru.wikipedia.org/wiki/Палочки_(игра)) на прямоугольном поле.

- Реализуйте игру между парой игроков.
- Реализуйте:
  - поддержку живых игроков
  - случайного игрока
  - игрока делающий ходы на основе какой-либо эвристической стратегии
- Реализуйте игру между произвольным числом игроков.
- Напишите программу, демонстрирующую работу вашей реализации.

Билет 40. Интерпретатор Spoon

Напишите интерпретатор языка программирования Spoon (http://ru.wikipedia.org/wiki/Spoon)

- Создайте программу `Spoon`, запускаемую с аргументами командной строки `program` и `input`.
- Ввод символов должен осуществляться из строки `input`, а вывод – на консоль.
- Должна быть возможность ограничить число действий, выполняемых программой и потребляемую память (если превышено – выводить сообщение об ошибке).

Билет 73. Протоколирование

Реализуйте набор классов для ведения протоколов.

- Создайте интерфейс `Logger` с методами:
  - `log(String message)`
  - `log(String message, Throwable cause)`
- Создайте три реализации: `FileLogger`, пишущий лог в текстовый файл; `ConsoleLogger`, записывающий лог на консоль и `CompositeLogger`, содержащий набор логгеров и передающий входящие вызовы им всем.
- Добавьте возможность задать специфичный формат записи лога. Например, включать в нее текущее время или дату.
- Добавьте возможность задать важность события: `Debug`, `Info`, `Warning` или `Error` и реализуйте возможность фильтровать сообщения, уровень важности которых не ниже заданного.
- Напишите программу, иллюстрирующую работу созданных классов.

Билет 16. Мат ладьей и королем

Реализуйте программу, ставящую мат пользователю в позиции вида «ладья и король против короля».

- *Исходное положение фигур задается в качестве аргументов командной строки.*
- *Программа играет за белых и делает первый ход.*
- *Пользователь играет за черных. Ходы вводятся в алгебраической нотации.*

Билет 29. Конъюнктивная нормальная форма

Преобразовать логическую формулу в конъюнктивную нормальную форму.

- Логическая формула может содержать:
  - константы: «0» — ложь и «1» — истина;
  - переменные: «a» … «z»;
  - логические связки: «|» — конъюнкция, «&» — дизъюнкция, «~» — отрицание;
  - скобки.
- *Логическая формула задается в виде аргумента командной строки, например:* `"(a & b)|~c"`.
- *Число различных переменных в формуле не превышает 10.*
- *Конъюнктивная нормальная форма формулы должна водиться на консоль. Для приведенного примера логической формулы результат может быть «(a|~c)&(b|~c)».*

Билет 21. Статистика по CSV

В качестве аргументов командой строки передается список файлов, которые надо обработать.

В каждом файле содержатся данные в формате CSV (https://en.wikipedia.org/wiki/Comma-separated_values) с заголовком. Для каждого столбца, указанного в заголовке нужно создать файл с именем `<название>.csv`, в котором вывести статистику по соответствующему столбцу. Статистика – значение и сколько раз оно встречается в данном столбце, в порядке убывания частоты.

Должны быть поддержаны следующие форматы (определяется расширением файла):

- `.csv`: разделитель «,», кавычка «"»;
- `.ssv`: разделитель «;», кавычка «'»;
- `.tsv`: разделитель символ табуляции, кавычка не поддерживается.

Билет 85. Система Лейтнера

Реализуйте интервальное повторение переводов слов на основе системы Лейтнера.

- Во входном файле даны пары слово (словосочетание) — перевод.
- Слова распределены на 10 корзин. Вероятность вытащить каждое слово из корзины n пропорциональна 1.5^n.
- Если пользователь ввел правильный перевод, то слово перемещается в корзину с номером на единицу больше (или остается в 10 корзине). Если пользователь ошибся, то слово перемещается в первую корзину.
- Если пользователь ввел пустой ответ, то программа должна завершиться, и при повторном запуске продолжить с того же места (с сохранением распределения слов по корзинам).

Билет 18. Кубик Рубика

Напишите программу, которая моделирует кубик Рубика.

- Исходно кубик собран (верхняя грань – белая (`W`), нижняя – желтая (`Y`) передняя – синяя (`B`), задняя – зеленая (`G`), левая – красная (`R`), правая – оранжевая (`O`)).
- Последовательность операций задается в виде аргумента командной тсроки в стандартной нотации (http://ru.wikipedia.org/wiki/Математика_кубика_Рубика), например, «`R'T`».
- *Развертка кубика после совершения всех операций должна быть выведена на консоль. Для примера, приведенного выше, программа должна вывести:*

```
   WWW
   WWW
   GGG
BBWOOOYGGRRR
RRRBBWOOOYGG
```

Билет 9. Реверси

Реализуйте игру “Реверси” (https://ru.wikipedia.org/wiki/Реверси) на прямоугольном поле.

- Реализуйте игру между парой игроков.
- Реализуйте:
  - поддержку живых игроков
  - случайного игрока
  - игрока делающий ходы на основе какой-либо эвристической стратегии
- Реализуйте проведение турнира между игроками, с выводом статистики.
- Напишите программу, демонстрирующую работу вашей реализации.

Билет 41. Проталкивание отрицаний

Преобразовать логическую формулу в форму, в которой отрицания применяются только к переменным, но не к константам и составным выражениям. Составные выражения упрощаются по формулам де Моргана и двойного отрицания.

- Логическая формула может содержать: константы: «0» — ложь и «1» — истина; переменные: «a» … «z»; логические связки: «|» — конъюнкция, «&» — дизъюнкция, «~» — отрицание; скобки.
- Логическая формула задается в виде аргумента командной строки, например: `"~(a & b)|~c"`.
- Преобразованная формула должна водиться на консоль. В формуле должны отсутствовать лишние скобки. Для приведенного примера логической формулы результат должен быть «~a|~b|~c».

Билет 92. Калах

Реализуйте игру "Калах" (https://ru.wikipedia.org/wiki/Калах)

- Реализуйте игру между парой игроков.
- Реализуйте:
  - поддержку живых игроков
  - случайного игрока
  - игрока делающий ходы на основе какой-либо эвристической стратегии
- Реализуйте проведение турнира между игроками, с выводом статистики.
- Напишите программу, демонстрирующую работу вашей реализации.

Билет 1. Автоматический переводчик

Программа читает входной файл, переводит его по словарю и пишет результат в выходной файл.

Формат словаря:

```
<слово или выражение> | <перевод>
```

Пример словаря

```
hello | здравствуй
friend | друг
I am | я
```

Примечания:

- При переводе регистр букв игнорируется.
- Если перевода нет в словаре – слово выводится без перевода.
- Если есть несколько подходящих вариантов перевода, выбирается вариант, с максимальной длиной левой части.

Билет 74. Дизъюнктивная нормальная форма

Преобразовать логическую формулу в дизъюнктивную нормальную форму.

- Логическая формула может содержать:
  - константы: «0» — ложь и «1» — истина;
  - переменные: «a» … «z»;
  - логические связки: «|» — конъюнкция, «&» — дизъюнкция, «~» — отрицание;
  - скобки.
- Число различных переменных в формуле не превышает 10.
- Логическая формула задается в виде аргумента командной строки, например: `"a & (b|~c)"`.
- Дизъюнктивная нормальная форма формулы должна водиться на консоль. Для приведенного примера логической формулы результат может быть «a&b|a&~c».

Билет 5. Шашки

Реализуйте игру "Шашки" (https://ru.wikipedia.org/wiki/Шашки) на доске 8×8 без дамок и последовательностей взятий.

- Реализуйте игру между парой игроков.
- Реализуйте:
  - поддержку живых игроков
  - случайного игрока
  - игрока делающий ходы на основе какой-либо эвристической стратегии
- Реализуйте проведение турнира между игроками, с выводом статистики.
- Напишите программу, демонстрирующую работу вашей реализации.

Билет 2. Sort

Напишите аналог утилиты `sort`.

- Поддерживаемые опции:
  - `--ignore-leading-blanks`
  - `--dictionary-order`
  - `--ignore-case`
  - `--ignore-nonprinting`
  - `--numeric-sort`
  - `--reverse`
- Ввод данных должен осуществляться с консоли или входного файла по выбору пользователя.

Билет 81. Grep

Напишите аналог утилиты `grep`.

- Поддерживаемые возможности:
  - Поиск нескольких слов, разделенных «|»
  - `--ignore-case`
- Данные, в которых осуществляется поиск могут не поместиться в памяти.
- Ввод данных должен осуществляться с консоли или входного файла по выбору пользователя.

Билет 42. bCodeToHTML (Clojure)

Реализуйте преобразовать из языка разметки bbCode (http://ru.wikipedia.org/wiki/BBCode).

- Создайте класс `Node`, описывающий элемент дерева разбора.
- Создайте подклассы `Node` для описания различных конструкций.
- Реализуйте разбор всех тегов.
- Создайте функцию `html`, выводящую разобранный текст в формате HTML (работает только в браузере).
- Сделайте программу, иллюстрирующую работу созданных классов.

Билет 87. bbCodeToHTML

Реализуйте преобразователь из языка разметки bbCode (http://ru.wikipedia.org/wiki/BBCode).

- Создайте класс `Node`, описывающий элемент дерева разбора.
- Создайте подклассы `Node` для описания различных конструкций.
- Реализуйте разбор тегов `[b]`, `[i]`, `[u]`, `[s]`, `[list]`.
- Создайте класс `HTMLWriter`, записывающий разобранный текст в HTML файл.
- Сделайте программу, читающую bbCode-файл и сохраняющую его в HTML-файл.

Билет 57. Судоку

Реализуйте решатель "Судоку" 3×3 (https://ru.wikipedia.org/wiki/Судоку)

- Входные данные должны читаться с консоли.
- Текущие результаты решения должны выводиться на консоль.
- Рекомендуется в начале перебирать клетки с минимальным числом оставшихся возможностей

Билет 82. Интерпретатор Brainfuck

Напишите интерпретатор языка программирования Brainfuck (http://ru.wikipedia.org/wiki/Brainfuck).

- Создайте программу `Brainfuck`, запускаемую с аргументами командной строки `program` и `input`.
- Ввод символов должен осуществляться из строки `input`, а вывод – на консоль.
- Должна быть возможность ограничить число действий, выполняемых программой и потребляемую память (если превышено – выводить сообщение об ошибке).

Билет 26. Шаблонизатор

Реализуйте простой шаблонизатор

- *На вход подаются два файла:* шаблон и подстановки
- *Требуется заменить в* шаблоне всех вхождения выражения вида «`$переменная`» *на соответствующую* подстановку
- *Подстановки задаются записями вида* «`переменная=значение`»
- *Если возможен выбор из нескольких* переменных, выбирается переменная *с самым длинным названием*
- *Название переменной может* содержать произвольные символы, кроме «=» *и перевода строки.*
- *Значение переменной может* содержать произвольные символы, кроме *перевода строки*

Билет 2. Устойчивая сортировка

Прочитать скрипт из входного файла и выполнить его. В скрипте встречаются следующие команды (каждая команда задается на отдельной строке):

- `add <index> <string>` — добавить строку с заданным индексом. Индексы — целые числа в диапазоне -10^9—10^9.
- `remove <index>` — удалить все строки с заданным индексом.
- `print <file>` — вывести в указанный файл строки, отсортированные по индексу, для строк с одинаковыми индексами — по времени добавления.

Формат вывода:

```
<index> <string>
```

При решении задачи запрещается использовать стандартный класс `Scanner`.