**Домашнее задание 1. 17.02.2013**

[Конспект пары](https://docs.google.com/document/d/1z4SMn57C3sJhe6f1iX0LRZi98iKqjmzwlZtPxSmLr2M/edit)  
[Презентация](https://docs.google.com/presentation/d/1y9JCQq4YbNSkd2FExRKRM9e9XCn6BZX1Zr7ovRs1_dM/edit)

1. Разработать интерфейс, представляющий структуру данных "связный список". Реализовать этот список двумя разными способами (односвязный/двусвязный, на массиве/на указателях и т.п.) на основе полученного интерфейса.
2. Разработать интерфейс, представляющий структуру данных "стек". На его основе реализовать стек двумя разными способами. Реализовать стековый калькулятор для подсчета арифметических выражений.

**Домашнее задание 2. 24.02.2014**

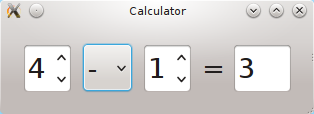
[Конспект пары](https://docs.google.com/document/d/1vknIkqYgd8V3EB_5Zvb38-7_YbjuWBSJ0RE3c6NV_xo/edit)  
[Презентация](https://docs.google.com/presentation/d/1dIZBYXf7BwJy_5bnaXrdohyuwckOcdeHZmNmfa_wGkc/edit)

1. Создать интерфейс сортировщика, принимающего массив элементов и выполняющего над ним сортировку. На его основе создать несколько классов, реализующих произвольные (разные) алгоритмы сортировки.
2. Дан массив размерностью N x N, N - нечетное число. Вывести элементы массива при обходе его по спирали, начиная с центра. Для решения задачи разработать интерфейс Выводилка с методом вывести(), реализовать на его основе два класса, осуществляющих либо вывод на консоль либо в файл. Написать программу, которая по желанию пользователя выбирает реализацию Выводилки и выводит массив.
3. Реализовать программу, сортирующую столбцы матрицы по первым элементам.

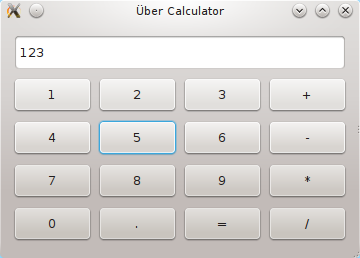
**Домашнее задание 3. 03.03.2014**

[Конспект пары](https://docs.google.com/document/d/1o3fJpkJIwutTejttVX_xRosyBQ91OKSw78mJwxASNNE/edit)

1. Создать графическое приложение с двумя виджетами — слайдером и прогрессбаром. Сделать так, чтобы при передвижении слайдера значение прогрессбара изменялось соответствующе.
2. Релизовать программу-калькулятор со следующим графическим интерфейсом (аргументы — QSpinBox, операция — QComboBox, значение — QLineEdit, при изменении значения аргумента или операции результат пересчитывается автоматически):



1. Реализовать продвинутый кнопочный калькулятор со следующим интерфейсом (для решения задачи использовать классы QGridLayout и QSignalMapper, взаимное расположение виджетов может быть другим):



**Домашнее задание 4. 10.03.2014**

[Конспект пары](https://docs.google.com/document/d/1vQs_RMJU80GpqpaVNwrgcszuPE9ydfRcqo2fORUIkGc/edit)  
[Презентация](https://docs.google.com/presentation/d/10PMtGDd9AWj_vRmaCr8xk6V7I7NNIgDr0Wi4jstM1PY/edit)

1. Реализовать класс для работы с хеш-таблицей (на списках). Общение с пользователем должно происходит в интерактивном режиме:  
   - добавить значение в хеш-таблицу  
   - удалить значение из хеш-таблицы  
   - поиск значения в хеш-таблице  
   - показать статистику по хеш-таблице (общее число ячеек, load factor, число конфликтов, максимальная длина списка в конфликтных ячейках и т.п.)  
   - выбрать хеш-функцию для подсчета хеша (из заранее заданных в коде)  
   Смена хэш-функции должна происходить во время работы программы, в класс используемая хеш-функция должна передаваться из клиентского кода.
2. Написать тесты ко всем предыдущим домашним задачам (к классам, отвечающим за GUI, тесты писать не нужно)

**Домашнее задание 5. 17.03.2014**

[Конспект пары](https://docs.google.com/document/d/1CraHG_JorgaTFu5Hy08uRCLIwcA_tpEzTXrDwKNHZfc/edit)  
[Презентация](https://docs.google.com/presentation/d/1COzhjz6gch872tWrT6Jg_9WJT-IeU-A2et39-KXd0js/edit)

1. Разработать и реализовать иерархию классов, описывающих дерево разбора арифметического выражения. Используя их, реализовать класс, вычисляющий значение выражения по дереву. Классы, представляющие операнды и операторы, должны сами уметь себя вычислять и печатать. Дерево разбора хранится в файле в виде (<операция> <операнд1> <операнд2>), где <операнд1> и <операнд2> сами могут быть деревьями, либо числами. Например, выражение (1 + 1) \* 2 представляется в виде (\* (+ 1 1) 2). Должны поддерживаться операции +, -, \*, / и целые числа в качестве аргументов. Требуется построить дерево в явном виде, распечатать его (не обязательно так же, как во входном файле) и посчитать значение выражения обходом дерева. Можно считать, что входной файл корректен. Пример — по входному файлу (\* (+ 1 1) 2) может печататься ( \* ( + 1 1 ) 2 ) и выводиться 4.
2. Создать шаблонный класс, реализующий АТД "Множество", обладающий операциями добавления, удаления элемента, проверки на принадлежность множеству, пересечения и объединения множеств.

**Домашнее задание 6. 24.03.2014**

[Конспект пары](https://docs.google.com/document/d/1QTLfJT1tUOdGF0HUPFK-NSu1RxlBeK03r9wnlEneegQ/edit)  
[Презентация](https://docs.google.com/presentation/d/1yfxO0cwphA10BapB-I7FEVA6XueRb2yydJK5429E3oc/edit)

1. Напишите программу, которая демонстрирует, что все деструкторы объектов, созданных в блоке, вызываются прежде, чем в этом блоке сгенерируется исключение.
2. На основе класса список из прошлой домашней работы реализовать класс UniqueList, который не содержит повторяющихся значений. Реализовать классы исключений, которые генерируются при попытке добавления в такой список уже существующего или при попытке удаления несуществующего элемента.
3. Реализовать шаблонный класс умного указателя SharedPointer, реализованного подсчетом ссылок на выделенную память (аналог std::shared\_ptr). Помимо прочего должен быть реализован конструктор копирования и оператор присваивания. Показать несколько примеров использования подобного класса.

**Домашнее задание 7. 07.04.2014**

[Конспект пары](https://docs.google.com/document/d/1JR7rRuLirx4xZFIIGU8ywowKvdoQbxkKu5Xsd9PhEW4/edit)  
[Презентация](https://docs.google.com/present/edit?id=0AU0yIfwI356AZGdqdHRyd21fMTQzZHRkN21mMjc)

1. Написать класс Bag, хранящий мультимножество (множество, в котором элементы могут повторяться). Класс должен быть ассимптотически эффективным по времени и по памяти (использовать самобалансирующиеся деревья для внутреннего представления).
2. Реализовать шаблонный класс Vector с методами (а где можно, операторами) сложения, вычитания, скалярного умножения и проверки на нулевой вектор. Вектор n-мерный. В классе должен быть реализован конструктор копирования.
3. Разработать приложение, позволяющие пользователю играть с самим собой в крестики-нолики. На экранной форме должно быть девять кнопок, расположенных в три столбца и три строки. При первоначальном нажатии на любую из кнопок, на ней появляется знак «Х». При дальнейшем нажатии на другую кнопку, на ней появляется знак «O». Повторное нажатие на кнопку не должно менять ее знака. Программа должна быть реализована таким образом, чтобы была возможность быстро поменять размер игрового поля.

**Домашнее задание 8. 17.04.2014**

[Презентация](https://docs.google.com/presentation/d/1tE5enB2wuCsLBh5VBp3iz7F0BE_HjeHT-15hKaMc2QU/edit?usp=sharing)

1. Реализовать два приложения, которые позволяют обмениваться друг с другом сообщениями (могут быть полезны стандартные примеры Qt Fortune Server Example и Fortune Client Example). Приложение server запускается и ждет входящих соединений на определенный порт, приложение client при запуске принимает номер порта сервера и соединяется с ним. Далее строки текста, вводимые в одном приложении, должны отображаться в другом. Графический интерфейс для приложений не обязателен.
2. Реализовать приложение с графическим интерфейсом, отображающее последние 10 цитат с bash.im. В случае разбора главной страницы могут быть полезны классы QWebPage, QWebElement и другие из модуля QtWebKit; в случае получения цитат из RSS (http://bash.im/rss/) могут быть полезны классы QDomDocument, QDomElement и другие из модуля QtXml.

**Домашнее задание 9. 21.04.2014**

[Репозиторий с кодом](https://github.com/jzuken/review2014)

1. Заревьюить чужой код. Код выдается персонально тем, кому осталось сдать менее десяти задач, и находится в репозитории по ссылке выше. Необходимо переложить нужный код в свой репозиторий и откомментировать его там средствами гитхаба.
2. Аналогично первой задаче, только код еще нужно и отладить. Результаты выложить в свой репозиторий.

Презентации с факультативных занятий

[Model-View и сцена в Qt](https://docs.google.com/presentation/d/1--DImmQa4LA3Be_lqX6vatGXwnP3i7AOxoCyVd1s9-I/edit?usp=sharing)  
[Базы данных](https://docs.google.com/presentation/d/1-w2cdLBgQVcFLcPvDMvaLSPpLZu_nhn8wjaKVCvy0i8/edit?usp=sharing)  
[Умные указатели](https://docs.google.com/presentation/d/1J-b40HyUZAwSrqVvknhLG-NGxOiWcOS18KFHwfSr45Y/edit?usp=sharing)

**Доклады**

[make и qmake](https://docs.google.com/presentation/d/1Pk7tSh4AJrq-GAdLTDL_xHs5y7vCYuTZ9WZ7tD45luw/edit?usp=sharing)  
[Лямбда выражения в стандарте C++11](https://docs.google.com/presentation/d/1rPd1k4PvqSzGVx-fe2Nfz9Jvh1wflDx-QtV30Ta2jmw/edit?usp=sharing)  
[QML и Qt Quick](https://docs.google.com/presentation/d/1_2GujkC-4tkGyHnMgBDKMx9u4Ajzv6_eReLo-x9h2mM/edit?usp=sharing)