



**Нижегородский государственный университет  
им. Н.И.Лобачевского**

***Факультет Вычислительной математики и кибернетики***

***Лабораторная работа 1***

***Разработка нового распределенного  
приложения для инструментария Alchemi***

## Слайд 1

---

**Г1**

1. Титульный слайд - согласовать с Гришагиным (образец Козинов)
2. Там, где слайд на заполнен - увеличить шрифт (см. слайд 2 - я его поправил)
3. На ряде слайдов сбивка шрифтов (в конце)
4. Слайд "Контрольные вопросы", а не "Вопросы для самопроверки"

Гергель; 02.09.2007

# Цель лабораторной работы

---

- ❑ Целью лабораторной работы является разработка *нового* распределенного приложения для инструментария Alchemi с использованием модели *грид-потоков*
- ❑ В качестве иллюстрирующего примера выступает задача визуализации функций комплексного переменного. Данная задача позволяет наглядно продемонстрировать все этапы разработки приложения, а также особенности его работы



# Упражнения

---

- ❑ *Упражнение 1*

Постановка задачи визуализации функции комплексного переменного

- ❑ *Упражнение 2*

Разработка алгоритма визуализации функции комплексного переменного

- ❑ *Упражнение 3*

Декомпозиция вычислительной схемы на независимые части

- ❑ *Упражнение 4*

Разработка распределенного приложения для инструментария Alchemi

- ❑ *Упражнение 5*

Получение сведений о выполняемом распределенном приложении



# Упражнение 1 – Постановка задачи...

- ❑ **Комплексное число** представляет собой пару действительных чисел  $(x, y)$ , где  $x$  - действительная часть комплексного числа, а  $y$  - его мнимая часть
- ❑ **Функцией комплексного переменного**  $w = f(z)$  называется правило, посредством которого всякому комплексному числу  $z$  из некоторого множества  $D$  ставится в соответствие комплексное число  $w$  из множества  $W$ . Множество  $D$  называется *областью определения* функции  $f(z)$ , а множество  $W$  – ее *областью значений*



# Упражнение 1 – Постановка задачи

---

- Пусть рассматривается некоторая функция комплексного переменного  $w = f(z)$ , заданная на прямоугольнике  $D = [a, b] \times [c, d]$ . Исходной информацией для задачи являются следующие данные:
  - символьное представление данной функции,
  - границы прямоугольника  $D$ , в котором будет проводиться визуализация
- Задача состоит в том, чтобы построить геометрический образ заданной функции в области  $D$  (метод излагается далее)



## Упражнение 2 – Метод визуализации...

- Рассмотрим функцию  $w = f(z)$ ,  $z = (x, y)$ ,  $w = (u, v)$
- Аргумент функции – комплексное число  $z$  – можно изобразить как точку на плоскости с координатами  $(x, y)$ . Значение  $w$  функции  $w = f(z)$  можно изобразить, например, с помощью цвета

⇒ Для этого каждому комплексному числу нужно поставить в соответствие уникальный цвет



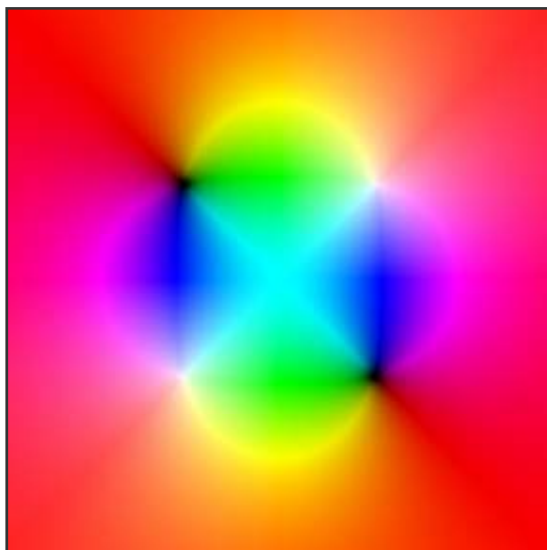
## Упражнение 2 – Метод визуализации...

- ❑ Комплексному числу 1 сопоставим **красный** цвет, двум другим кубичным корням из единицы –  $e^{2\pi i/3}$  и  $e^{-2\pi i/3}$  – **зеленый** и **синий** соответственно. Цвет любого другого значения аргумента, определяется линейной интерполяцией
- ❑ Чтобы учесть *модуль* комплексного числа, ноль окрасим **белым** цветом, а бесконечно удаленную точку – **черным**. Любое значение модуля комплексного числа получит уникальный оттенок серого цвета

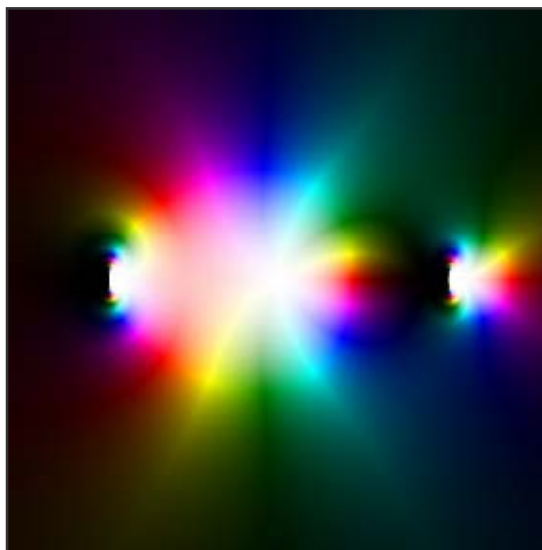


## Упражнение 2 – Метод визуализации

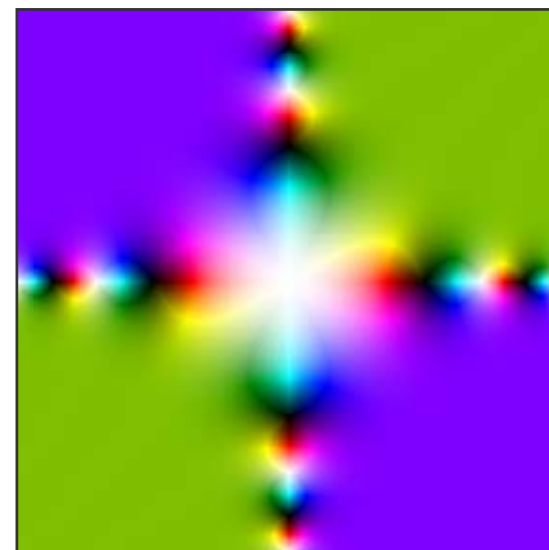
- Данный метод позволяет: определить особые точки и нули, установить кратность нулей и полюсов, определить области однолистности, охарактеризовать общее поведение функции
- Визуализация больших изображений для сложных функций является трудоемкой вычислительной задачей



$$w = (z^2 - i) / (z^2 + i)$$



$$w = z^2 \cdot e^{tg(z)}$$



$$w = tg(z^2)$$

## Упражнение 3 – Декомпозиция вычислений...

- ❑ Напомним основные этапы решения вычислительной задачи с помощью инструментария Alchemi:
  - произвести декомпозицию вычислительной схемы на *независимые* подзадачи;
  - реализовать *грид-потоки* для обработки данных подзадач;
  - организовать выполнение грид-потоков в вычислительной сети с помощью класса *грид-приложения*
- ❑ Рассмотрим первый этап решения задачи



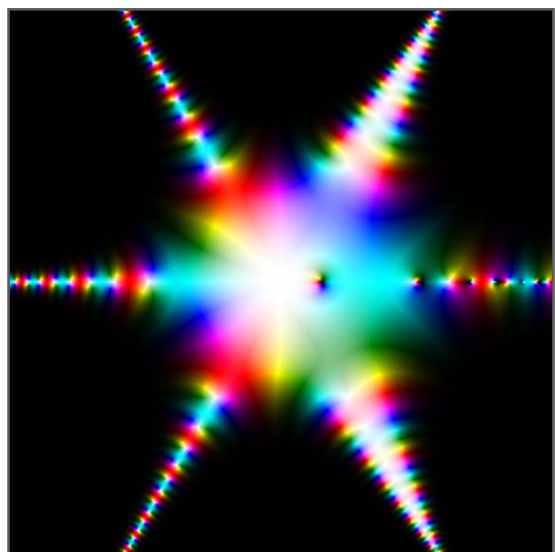
## Упражнение 3 – Декомпозиция вычислений...

- При использовании инструментария Alchemi можно выделить два основных требования, которым должен удовлетворять выбираемый подход:
  - обеспечение примерно *равного* объема вычислений в выделяемых подзадачах;
  - *отсутствие* информационных зависимостей между этими подзадачами
- Рассмотрим два часто встречающихся типа вычислительных схем, применимых, в том числе, и для инструментария Alchemi:
  - вычисления сводятся к выполнению *однотипной* обработке элементов большого набора данных. В таких задачах существует *параллелизм по данным*, и выделение подзадач сводится к разделению имеющихся данных;
  - вычисления могут состоять в выполнении *разных* операций над одним и тем же набором данных – в этом случае имеет место *функциональный параллелизм*

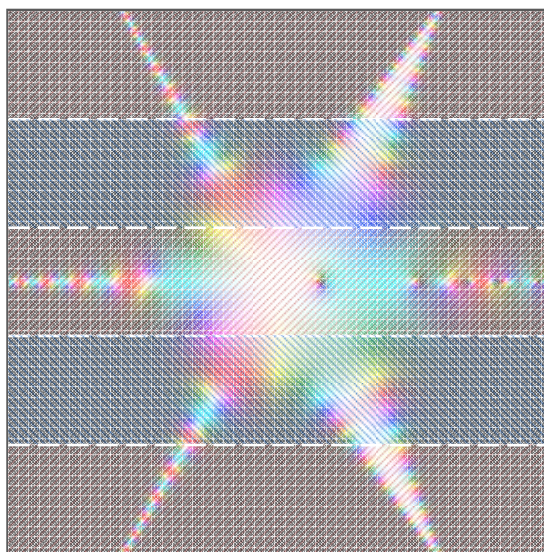


## Упражнение 3 – Декомпозиция вычислений...

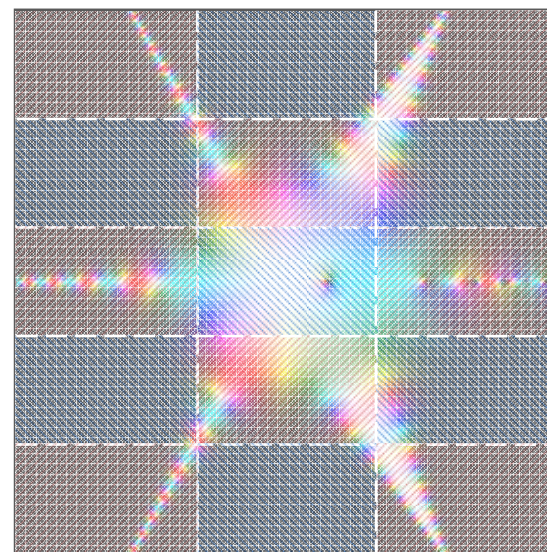
- В нашей учебной задаче очевиден параллелизм по данным: исходное множество пикселей генерируемого изображения может быть разделено на отдельные группы строк – *ленточная схема* разделения данных или на прямоугольные наборы пикселей – *блочная схема* разделения данных



Исходное изображение



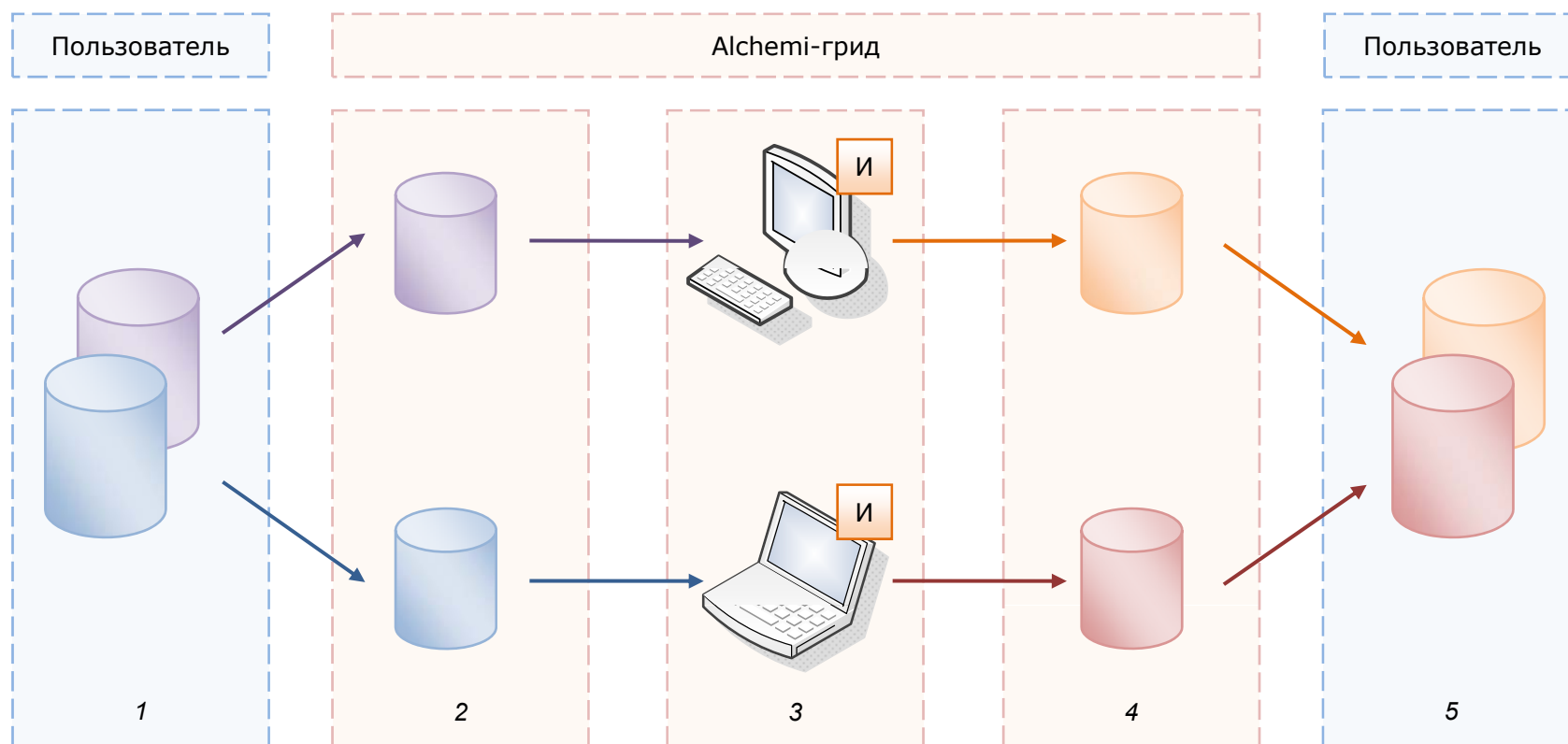
Ленточная схема



Блочная схема

## Упражнение 3 – Декомпозиция вычислений...

- Рассмотрим процесс распределенной обработки данных в Alchemi-грид



## Упражнение 3 – Декомпозиция вычислений

- При декомпозиции вычислений всегда следует учитывать накладные *расходы на передачу грид-поток*ов на удаленные машины (может достигать нескольких секунд!), а также общее *число доступных Исполнителей*:
  - объем вычислений, производимый одним грид-потоком, не должен быть слишком мал (время вычислений >> время передачи);
  - общее число грид-потоков не должно в десятки или даже сотни раз превышать число доступных Исполнителей



# Упражнение 4 – Разработка грид-приложения

- ❑ *Задание 1*  
Открытие проекта **ComplexVisual**
- ❑ *Задание 2*  
Разработка класса грид-потока
- ❑ *Задание 3*  
Запуск грид-потоков на локальной машине
- ❑ *Задание 4*  
Тестирование приложения на локальной машине
- ❑ *Задание 5*  
Запуск грид-потоков в вычислительной грид
- ❑ *Задание 6*  
Тестирование приложения в вычислительной грид

Универсальная схема



# Задание 1 – Открытие проекта ComplexVisual

- ❑ Для открытия проекта **ComplexVisual** выполните следующие шаги:
  - запустите среду Microsoft Visual Studio 2005, если она еще не запущена,
  - в меню **File** выполните команду **Open | Project/Solution**,
  - в диалоговом окне **Open Project** выберите папку **ComplexVisual**,
  - выберите файл **ComplexVisual.sln** и выполните команду **Open**
- ❑ Проект включает в себя:
  - библиотеку **MathTools** – содержит вспомогательные математические классы, которые используются основным приложением (классы комплексных чисел, функций и синтаксический анализатор);
  - библиотеку **ComplexThread** – содержит реализацию класса грид-потока;
  - проект **AlchemiApplication** – основное приложение, в котором формируются грид-потоки и отправляются для исполнения в вычислительную грид, построенную на базе инструментария Alchemi



## Задание 2 – Разработка класса грид-потока...

- ❑ Чтобы открыть заготовку кода для класса грид-потока разверните проект **ComplexThread** и дважды щелкните мышкой на файле **ComplexThread.cs**  *Открыть*
- ❑ Замечания об объявлении класса грид-потока:
  - перед объявлением класса необходимо подключить библиотеки **MathTools** и **Alchemi** (последняя подключается в *любом* проекте);
  - класс грид-потока наследуется от *абстрактного* класса **GThread**, в котором определяется базовая функциональность для всех грид-потоков;
  - перед объявлением класса необходимо поместить атрибут **Serializable** для того, чтобы объект был снабжен методами сериализации и десериализации



## Задание 2 – Разработка класса грид-потока

- ❑ Основные этапы реализации класса грид-потока:
  - определить все переменные, которые могут потребоваться для *вычислений* и последующего *восстановления результата* на машине пользователя из имеющихся фрагментов, вычисленных отдельными грид-потоками  *Открыть*
  - создать *конструктор* для класса грид-потока. Действия конструктора, как правило, сводятся к инициализации полей класса и выделению необходимых ресурсов  *Открыть*
  - реализовать *основной* метод грид-потока, в котором будет производиться обработка заданной порции данных – метод **void Start()**. В каждой конкретной реализации данный метод необходимо переопределять, заменяя ключевое слово **abstract** словом **override**  *Открыть*
  - для того чтобы получить доступ к необходимым полям класса и при этом не нарушить инкапсуляцию данных, в конце класса определить *свойства* для доступа к данным полям  *Открыть*



## Задание 3 – Локальный запуск грид-потоков...

- ❑ Прежде чем организовывать исполнение грид-потоков в Alchemi-грид желательно произвести тестирование вычислительного кода на локальной машине. Выявленные на этом этапе ошибки будет проще устранить
- ❑ Формирование грид-потоков и организация их исполнения осуществляется в основном приложении **ComplexVisual**. Для того чтобы открыть заготовку кода выберите в окне **Solution Explorer** проект **ComplexVisual**, затем правой кнопкой мышки щелкните на файле **MainForm.cs** и в появившемся контекстном меню выберите команду **View Code**  *Открыть*
- ❑ Обратите внимание, что перед объявлением класса необходимо подключить библиотеки **MathTools** и **Alchemi** (последняя подключается в любом проекте)



## Задание 3 – Локальный запуск грид-потоков...

- ❑ Возможная схема реализации запуска грид-потоков на локальной машине:
  - определить все глобальные переменные, которые могут потребоваться *во время вычислений* и для *сохранения результатов*  [Открыть](#)
  - реализовать метод, в котором осуществляется обработка грид-потоков на локальной машине. В нашем приложении данный метод называется **void BuildGraphLocal()**  [Открыть](#)
  - Реализовать метод для обновления прогресса вычислений (различные действия, связанные с пользовательским интерфейсом – обновление полосы прогресса, какого-либо изображения, текста с затраченным временем и т.д.). В нашем приложении данный метод называется **void UpdateProgress(GThread thread)**  [Открыть](#)



## Задание 3 – Локальный запуск грид-потоков

---

- Замечания о локальном запуске грид-потоков:
  - использование нескольких грид-потоков в данном случае не приводит к какому-либо ускорению вычислений. Это связано с тем, что все грид-потоки выполняют свою работу последовательно, а не одновременно. Следующий грид-поток запускается лишь тогда, когда предыдущий завершает свою работу;
  - однако такой подход позволяет быстро и просто отладить приложение и с минимальными усилиями реализовать обработку грид-потоков в вычислительной сети



## Задание 4 – Тестирование приложения

- ❑ Для того чтобы откомпилировать и запустить приложение выполните команду **Debug | Start Debugging** или нажмите клавишу **F5**
- ❑ Для ввода параметров служат следующие группы:
  - **Resolution**: поля ввода для установки размеров генерируемого изображения (ширина и высота);
  - **Cells**: поля ввода для числа горизонтальных и вертикальных разбиений изображения, влияющих на общее число грид-потокков;
  - **Ranges**: поля ввода для установки границ прямоугольной области, в которой будет визуализироваться комплексная функция;
  - **Function**: единственное поле ввода для задания аналитического выражения комплексной функции;
  - **Color**: поля ввода для установки дополнительных параметров, задающих насыщенность белого и черного цвета в генерируемом изображении

*🖱 Заполнив поля ввода, щелкните на кнопке **Build Graph on Local***



## Задание 5 – Запуск грид-поток в грид...

- Замечания о библиотеке Windows Forms и инструментарии Alchemi:
  - все необходимые сервисные операции инструментарий Alchemi выполняет в *отдельном* рабочем потоке, периодически посылая уведомления в основной поток приложения. Уведомления могут посылаться при успешном или неудачном завершении грид-потока, а также при завершении работы грид-приложения;
  - библиотека пользовательского интерфейса Windows Forms работает с окном и его элементами управления в *основном* потоке. Из “внешнего” потока невозможно получить доступ к окну и его элементам управления;
  - для реализации взаимодействия необходимо использовать средства синхронизации, предоставляемые данной библиотекой. Одним из таких средств является метод базового класса **Control**:

```
public virtual IAsyncResult BeginInvoke(Delegate, object[]);
```



## Задание 5 – Запуск грид-потоков в грид

- ❑ Возможная схема реализации запуска грид-потоков в Alchemi-грид:
  - определить все глобальные переменные, которые могут потребоваться *во время вычислений* и для *сохранения результатов*  [Открыть](#)
  - реализовать метод, в котором осуществляется обработка грид-потоков в грид. В нашем приложении данный метод называется **void BuildGraphGrid()**  [Открыть](#)
  - реализовать обработчики событий **ThreadFinish** и **ThreadFailed**, которые вызываются при успешном и неудачном (в процессе работы возникло исключение) завершении грид-потока соответственно  [Открыть](#)
  - удалить экземпляр грид-приложения при выходе из программы, иначе на Менеджере останутся связанные с ним ресурсы  [Открыть](#)



## Задание 6 – Тестирование приложения

---

- ❑ Для того чтобы откомпилировать и запустить приложение выполните команду **Debug | Start Debugging** или нажмите клавишу **F5**
- ❑ На машине разработчика следует создать минимальную “грид” из одного Менеджера и одного Исполнителя. При наличии многоядерной или многопроцессорной машины желательно запускать несколько Исполнителей для более оптимального использования вычислительных ресурсов. Если имеется возможность, отдельные Исполнители лучше запустить на различных компьютерах сети

*✎ Заполнив поля ввода, щелкните на кнопке **Build Graph on Grid***



## Упражнение 5 – Мониторинг приложений...

- ❑ Инструментарий Alchemi включает средства мониторинга ключевых параметров быстродействия вычислительной грид и запущенных приложений. Для этой цели служит приложение Alchemi Console, которое входит в стандартную поставку Alchemi SDK
- ❑ Для запуска программы Alchemi Console перейдите в каталог, в который был распакован Alchemi SDK, откройте подкаталог **bin** и запустите файл **Alchemi.Console.exe**
- ❑ Подключитесь к Alchemi-грид с помощью кнопки на панели инструментов, отвечающей за установку соединения (крайняя слева)

*➤ После успешного подключения можно приступать к работе*



## Упражнение 5 – Мониторинг приложений

---

- ❑ Все действия, производимые с помощью приложения Alchemi Console, разделены на категории и представлены в виде иерархической структуры в левой части окна:
  - **Executors**: просмотр списка доступных Исполнителей. Для получения более подробных сведений выберите нужный Исполнитель и дважды щелкните по нему левой кнопкой мышки;
  - **Applications**: просмотр списка приложений, которые были запущены в Alchemi-грид. Для более подробной информации о некотором приложении выберите его левой кнопкой мышки
- ❑ В тех случаях, когда требуется получить информацию о быстродействии вычислительной грид в целом, можно воспользоваться окном **System Performance Monitor**. Для вызова данного окна служит кнопка на панели инструментов (крайняя справа)



# Заключение...

---

- ❑ Инструментарий Alchemi предлагает довольно обширные и простые в использовании возможности для разработки грид-приложений
- ❑ К преимуществам инструментария Alchemi можно отнести:
  - простота разработки грид-приложений и компактность кода;
  - возможность быстро и удобно проектировать пользовательский интерфейс;
  - легкость в отладке и использовании приложений;
  - возможность запуска разработанных приложений на локальных компьютерах без установленных компонент инструментария Alchemi



# Заключение

---

- ❑ К недостаткам инструментария Alchemi можно отнести:
  - отсутствие переносимости кода и ориентация на единственную платформу;
  - ограниченный класс задач из-за отсутствия обмена данными между грид-потоками;
  - слабая масштабируемость инструментария



# Вопросы для самопроверки...

---

1. Какие модели программирования предоставляет инструментарий Alchemi? В чем их особенности?
2. В каких ситуациях следует применять модель грид-поток?
3. В чем состоят основные этапы разработки приложения с помощью модели грид-поток?
4. Каким требованиям должна удовлетворять вычислительная схема?
5. Опишите два стандартных вида параллелизма.
6. Опишите два стандартных вида вычислительных схем.
7. В чем состоят основные этапы разработки класса грид-поток?
8. Каким образом лучше всего отлаживать класс грид-поток?



# Вопросы для самопроверки

---

9. Каким образом осуществляется запуск грид-потоков в вычислительной сети?
10. Какие средства предоставляет инструментарий Alchemi для мониторинга грид-приложений?



# Дополнительные упражнения

---

1. Добавить возможность остановки вычислений при их запуске как на локальном компьютере, так и в вычислительной грид
2. Реализовать корректную обработку неудачно завершившихся грид-поток и последующий их перезапуск на доступных Исполнителях (если это возможно)
3. В качестве другого дополнительного упражнения можно предложить модифицировать программу таким образом, чтобы она строила различные фрактальные множества, такие как множество Мандельброта и Джулии (подобная модификация не займет много времени). Подробные сведения о фрактальных множествах можно найти в работе [2]



# Литература

---

1. Ахо, Альфред, В., Сети, Рави, Ульман, Джеффри, Д. Компиляторы: принципы, технологии и инструменты.: Пер. с англ. – М.: Издательский дом “Вильямс”, 2003. – 768 с.
2. Никулин Е. А. Компьютерная геометрия и алгоритмы машинной графики. – СПб.: БХВ-Петербург, 2003. – 560 с.
3. Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal. Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework.  
[http://www.gridbus.org/~alchemi/files/alchemi\\_bookchapter.pdf](http://www.gridbus.org/~alchemi/files/alchemi_bookchapter.pdf)
4. Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal. Alchemi: A .NET-based Enterprise Grid Computing System.  
[http://www.gridbus.org/%7Eraj/papers/alchemi\\_icomp05.pdf](http://www.gridbus.org/%7Eraj/papers/alchemi_icomp05.pdf)



# Литература

---

5. Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal. Alchemi: A .NET-based Grid Computing Framework and its Integration into Global Grids.  
[http://www.gridbus.org/~alchemi/files/alchemi\\_techreport.pdf](http://www.gridbus.org/~alchemi/files/alchemi_techreport.pdf)
6. <http://www.alchemi.net> – официальный сайт проекта Alchemi

