

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное учреждение
высшего профессионального образования
"Казанский (Приволжский) федеральный университет"
Химический институт им. А.М. Бутлерова



УТВЕРЖДАЮ

Проректор
по образовательной деятельности КФУ
Проф. Минзарипов Р.Г.

"__" _____ 20__ г.

Программа дисциплины

Управление программным обеспечением М2.В.5

Направление подготовки: 020100.68 - Химия

Профиль подготовки: Хемоинформатика и молекулярное моделирование

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Андрианова А.А. , Миниахметов Э.Р.

Рецензент(ы):

Миссаров М.Д.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Латыпов Р. Х.

Протокол заседания кафедры No ____ от "____" _____ 201__ г

Учебно-методическая комиссия Химического института им. А.М. Бутлерова:

Протокол заседания УМК No ____ от "____" _____ 201__ г

Регистрационный No

Казань
2013

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) доцент, к.н. Андрианова А.А. кафедра системного анализа и информационных технологий отделение фундаментальной информатики и информационных технологий , Anastasiya.Andrianova@kpfu.ru ; ассистент, б/с Миниахметов Э.Р. кафедра анализа данных и исследования операций отделение фундаментальной информатики и информационных технологий

1. Цели освоения дисциплины

Целью освоения дисциплины "Управление программным обеспечением" является выработка у студентов знаний, каким образом необходимо создавать быстро работающий и легкий для доработки программный код. Эти знания необходимы для разработки программного кода в команде из нескольких программистов, создания программ, удобных для использования сторонними пользователями, легкой разработки и доработки приложения через некоторое время после его создания, создании кода, легко "читаемого" сторонними разработчиками. Создание работоспособных программ и скриптов, которые можно легко использовать и дорабатывать является важной задачей для специалиста в области хемоинформатики и молекулярного моделирования, поскольку распространение работоспособных и эффективных моделей в научном сообществе зачастую происходит за счет обмена готовыми программами.

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел " М2.В.5 Профессиональный" основной образовательной программы 020100.68 Химия и относится к вариативной части. Осваивается на 1 курсе, 2 семестр.

Дисциплина "Управление программным обеспечением" тесно связана с другими дисциплинами информационного профиля данной магистратуры. Для ее освоения необходимы знания, приобретаемые студентами в ходе изучения дисциплин "Основы компьютерного программирования в приложении к химическим задачам", "Компьютеры, операционные системы и сети", "Программирование на C, C++ и Java", "Web-технологии в химии". Изучение данной дисциплины необходимо для успешного выполнения квалификационной работы, работы в рамках научно-производственной практики, а также изучения дисциплин "Хемоинформатика" и "Биоинформатика".

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОК-2 (общекультурные компетенции)	умением принимать нестандартные решения
ОК-3 (общекультурные компетенции)	владением иностранным (прежде всего английским) языком в области профессиональной деятельности и межличностного общения
ОК-5 (общекультурные компетенции)	владением современными компьютерными технологиями, применяемыми при обработке результатов научных экспериментов и сборе, обработке, хранении и передаче информации при проведении самостоятельных научных исследований
ПК-10 (профессиональные компетенции)	способностью определять и анализировать проблемы , планировать стратегию их решения

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-5 (профессиональные компетенции)	способностью анализировать полученные результаты, делать необходимые выводы формулировать предложения

В результате освоения дисциплины студент:

1. должен знать:

- основные способы и приемы улучшения программного кода,
- типичные ошибки, допускаемые при программировании, которые приводят к замедлению работы или ухудшению читаемости кода.

2. должен уметь:

- оптимизировать программный код, улучшать его читаемость, проводить дебаггинг,
- создавать версии программного кода при работе над ним,
- создавать программный код в команде разработчиков,
- создавать программный код, допускающий его легкое понимание, редактирование, доработку и работу с ним,
- настраивать возможность корректной инсталляции программы на операционные системы

3. должен владеть:

приемами дебаггинга, рефакторинга, версионирования и оптимизации программного кода

проводить улучшение программного кода для улучшения скорости его работы, читаемости и легкости редактирования

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет зачетных(ые) единиц(ы) 72 часа(ов).

Форма промежуточного контроля дисциплины зачет во 2 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Основные проблемы и ошибки при написании программного кода	2	1	2	0	0	

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
2.	Тема 2. Написание корректного программного кода. Версионирование	2	2-4	6	0	0	домашнее задание
3.	Тема 3. Рефакторинг кода	2	5-7	6	0	0	домашнее задание
4.	Тема 4. Оптимизация кода	2	8-9	4	0	0	домашнее задание
5.	Тема 5. Дебаггинг	2	10-12	4	0	0	домашнее задание
6.	Тема 6. Компиляция и установка программ	2	13-14	4	0	0	домашнее задание
7.	Тема 7. Разработка программы в команде	2	15	2	0	0	
	Тема . Итоговая форма контроля	2		0	0	0	зачет
	Итого			28	0	0	

4.2 Содержание дисциплины

Тема 1. Основные проблемы и ошибки при написании программного кода

лекционное занятие (2 часа(ов)):

Основные проблемы и ошибки при написании программного кода. Примеры плохо читаемого и неоптимизированного кода: дублирование кода, длинный метод, большой класс, "ленивый" класс, длинный список параметров, "завистливые" функции, избыточные временные переменные, классы данных, несгруппированные данные, плохие имена функций.

Реинжиниринг. Проблемы, возникающие при реинжиниринге.

Тема 2. Написание корректного программного кода. Версионирование

лекционное занятие (6 часа(ов)):

Написание корректного программного кода. Типичный алгоритм написания программы. Основные приемы написания программ, применяющиеся при их создании. Комментирование. Корректное создание классов и функций. Корректные имена классов и функций. Создание библиотек - когда нужно и когда вредно. Приемы работы над работоспособным кодом. Версионирование. Приемы версионирования программы. "Откаты" к предыдущим рабочим версиям. Обнаружение ошибок. Бенчмаркинг программы. Альфа-тестирование.

Тема 3. Рефакторинг кода

лекционное занятие (6 часа(ов)):

Цели рефакторинга. Причины применения рефакторинга. Признаки плохого кода. Рефакторинг кода. Методы рефакторинга. Изменение сигнатуры метода (Change Method Signature). Инкапсуляция поля (Encapsulate Field). Выделение класса (Extract Class). Выделение интерфейса (Extract Interface). Выделение локальной переменной (Extract Local Variable). Выделение метода (Extract Method). Генерализация типа (Generalize Type). Встраивание (Inline). Введение фабрики (Introduce Factory). Введение параметра (Introduce Parameter). Подъём метода (Pull Up Method). Спуск метода (Push Down Method). Переименование метода (Rename Method). Перемещение метода (Move Method). Замена условного оператора полиморфизмом (Replace Conditional with Polymorphism). Замена наследования делегированием (Replace Inheritance with Delegation). Замена кода типа подклассами (Replace Type Code with Subclasses). Проблемы, возникающие при проведении рефакторинга. Средства автоматизации рефакторинга.

Тема 4. Оптимизация кода

лекционное занятие (4 часа(ов)):

Оптимизация. Алгоритм. Прототип. Принцип Парето. Основы оптимизации. Компромиссы (tradeoff). Различные области. Узкие места. Простейшие приёмы оптимизации программ по затратам процессорного времени. Инициализация объектов данных. Программирование арифметических операций. Циклы. Инвариантные фрагменты кода. Отложенные вычисления. Мемоизация. Оптимизирующий компилятор. Основы параллельного выполнения кода на нескольких процессорах.

Тема 5. Дебаггинг

лекционное занятие (4 часа(ов)):

Отладка. Назначение и способы. Место отладки в цикле разработки программы. Инструменты отладки. Профилировщики. API логгеры. Дизассемблеры. Снифферы. Снифферы аппаратных интерфейсов. Логи. Инструменты, снижающие потребность в отладке. Контрактное программирование. Модульное тестирование. Статический анализ кода. Безопасность программного кода и отладка. SQL-инъекция и переполнение буфера. Выявление недокументированного поведения системы. Устранение небезопасного кода. Статический анализ кода. Фаззинг.

Тема 6. Компиляция и установка программ

лекционное занятие (4 часа(ов)):

Компиляция и установка программ. Виды компиляторов. Виды компиляции. Структура компилятора. Генерация кода. Динамическая компиляция. Раздельная компиляция. Компиляция сложных программ. Make скрипты.

Тема 7. Разработка программы в команде

лекционное занятие (2 часа(ов)):

Разработка программы в команде. Основные приемы программирования в команде. Программы и системы для облегчения разработки в команде.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
2.	Тема 2. Написание корректного программного кода. Версионирование	2	2-4	подготовка домашнего задания	8	домашнее задание
3.	Тема 3. Рефакторинг кода	2	5-7	подготовка домашнего задания	10	домашнее задание

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
4.	Тема 4. Оптимизация кода	2	8-9	подготовка домашнего задания	8	домашнее задание
5.	Тема 5. Дебаггинг	2	10-12	подготовка домашнего задания	8	домашнее задание
6.	Тема 6. Компиляция и установка программ	2	13-14	подготовка домашнего задания	10	домашнее задание
	Итого				44	

5. Образовательные технологии, включая интерактивные формы обучения

Дисциплина представляет собой цикл лекций, совмещенных с выполнением домашней работы с использованием знаний, полученных на лекциях. Домашние задания посвящены выработке базовых навыков модификации программ с целью улучшения их работы. Домашние задания могут выполняться в компьютерных классах с использованием компьютеров и предустановленного программного обеспечения.

В лекционных занятиях используются следующие технологии интерактивного обучения:

- проблемно-ориентированная беседа на отдельные темы
- дискуссии со студентами по проблемам приведенного кода и возможным решениям проблемы.

В качестве зачета учащиеся должны самостоятельно создать в команде собственную программу по заданию преподавателя. Каждый учащийся должен создать собственный модуль программы. Сложность задания состоит в необходимости создания многомодульной программы без наличия ошибок, с необходимыми пояснениями, с использованием знаний, полученных в рамках курса "Управление программным обеспечением".

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. Основные проблемы и ошибки при написании программного кода

Тема 2. Написание корректного программного кода. Версионирование

домашнее задание , примерные вопросы:

Заданное преподавателем модифицирование программного кода, написанного на языке C. Создание кода с атрибутами версионирования.

Тема 3. Рефакторинг кода

домашнее задание , примерные вопросы:

Рефакторинг заданного программного кода, написанного на языке C. Обнаружение и устранение проблем.

Тема 4. Оптимизация кода

домашнее задание , примерные вопросы:

Оптимизация заданного программного кода, написанного на языке C. Обнаружение проблем кода и их устранение.

Тема 5. Дебаггинг

домашнее задание , примерные вопросы:

Отладка заданного программного кода, написанного на языке C и содержащего критическую ошибку. Обнаружение положения ошибки и ее устранение.

Тема 6. Компиляция и установка программ

домашнее задание , примерные вопросы:

Создание make-скрипта для заданной сложной программы.

Тема 7. Разработка программы в команде

Тема . Итоговая форма контроля

Примерные вопросы к зачету:

Вопросы к зачету:

1. Основные проблемы и ошибки при написании программного кода
2. Реинжиниринг. Проблемы, возникающие при реинжиниринге.
3. Типичный алгоритм написания программы. Основные приемы написания программ.
4. Версионирование. Приемы версионирования программы.
5. Рефакторинг кода. Причины применения рефакторинга. Признаки плохого кода.
6. Методы рефакторинга.
7. Основы оптимизации кода, компромиссы и узкие места.
8. Простейшие приёмы оптимизации программ по затратам процессорного времени.
9. Дебаггинг (отладка). Место отладки в цикле разработки программы.
10. Инструменты отладки.
11. Компиляция и установка программ. Виды компиляторов. Виды компиляции. Структура компилятора.
12. Make скрипты. Назначение и основные команды.
13. Основы программирования в команде.

7.1. Основная литература:

1. Андрианова, А.А. Практикум по курсу "Алгоритмизация и программирование". Часть 1 / А.А. Андрианова, Т.М. Мухтарова. - Казань: Казанский государственный университет, 2008. - 96 с.
2. Андрианова, А.А. Практикум по курсу "Алгоритмизация и программирование". Часть 2. / А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. - Казань: Казанский государственный университет, 2009. - 132 с.
3. Андрианова, А.А. Объектно-ориентированное программирование на C++. Часть 1. / А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. - Казань: Казанский федеральный университет, 2010. - 124 с.
4. Андрианова, А.А. Объектно-ориентированное программирование на C++. Часть 2. / А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. - Казань: Казанский федеральный университет, 2010. - 116 с.
5. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации / А.П. Пятибратов, Л.П. Гудыно, А.А. Кириченко. - М: Финансы и статистика, 2008. - 512 с.

7.2. Дополнительная литература:

1. М. Фаулер. Рефакторинг. Улучшение существующего кода: в пер. С. Макавеева. М.: Символ-плюс. 2008. 432 с.
2. Крис Касперски. Техника оптимизации программ. Эффективное использование памяти. СПб.: БХВ-Петербург, 2003. -464 с.
3. Дональд Кнут. Искусство программирования = The Art of Computer Programming. 3-е изд. М.: Вильямс, 2006. - Т. 1: Основные алгоритмы. ? 720 с.
4. Дональд Кнут. Искусство программирования = The Art of Computer Programming. 3-е изд. М.: Вильямс, 2007. - Т. 2: Получисленные алгоритмы. ? 832 с.

5. Дональд Кнут. Искусство программирования = The Art of Computer Programming. 2-е изд. М.: Вильямс, 2007. - Т. 3: Сортировка и поиск. ? 824 с.
6. Шилдт, Г. Самоучитель C++: Пер. с англ. / Герберт Шилдт. ?3-е изд., перераб. и доп.-Санкт-Петербург: БХВ-Петербург, 2003. ?683с.
7. Бежанова, М. М. Практическое программирование: структуры данных и алгоритмы: учебник / М. М. Бежанова, Л. А. Москвина, И. В. Поттосин.-Москва: Логос, 2001. -223 с.
8. Мартин, Роберт К. Быстрая разработка программ: Принципы, примеры, практика / Р.К. Мартин, Д.В. Ньюкирк, Р.С. Косс; [Пер. с англ.: А.П. Сергеев, Т.А. Шамренко; Под ред. А.П.Сергеева].-М. и др.: Вильямс, 2004.-739 с.

7.3. Интернет-ресурсы:

Home of the MinGW and MSYS Projects - <http://www.mingw.org/>

Wikipedia. Категория:Оптимизация программного обеспечения -

[illegible]

Рефакторинг - посты на Habrahabr.ru -

[http://habrahabr.ru/search/?q=\[%D1%80%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%8](http://habrahabr.ru/search/?q=[%D1%80%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%8)

Рефакторинг (C#) - <http://msdn.microsoft.com/ru-ru/library/719exd8s.aspx>

Управление программным обеспечением -

<http://www.openspin.org/materials/courses/admin/ch03s02.html>

8. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Освоение дисциплины "Управление программным обеспечением" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 020100.68 "Химия" и магистерской программе Хемоинформатика и молекулярное моделирование .

Автор(ы):

Андрианова А.А. _____

Миниахметов Э.Р. _____

"__" _____ 201__ г.

Рецензент(ы):

Миссаров М.Д. _____

"__" _____ 201__ г.

Лист согласования

N	ФИО	Согласование
1	Латыпов Р. Х.	
2	Антипин И. С.	
3	Бычкова Т. И.	
4	Чижанова Е. А.	
5	Соколова Е. А.	
6	Тимофеева О. А.	