

Министерство образования и науки Российской Федерации
ФГБОУ ВО «МГУТУ им. К. Г. Разумовского (Первый казачий университет)»
Университетский колледж информационных технологий

Рассмотрено предметной (цикловой) комиссией специальности «Программирование в компьютерных системах»

Протокол №__

«__» _____ 2016 г.

Председатель _____ /Кириллов А. И./

Утверждаю

зам. директора по УМР

«__» _____ 2016 г.

_____/_____/

Вопросы для повторения к зачёту по МДК.03.02

«Инструментальные средства разработки программного обеспечения»

Специальность: 09.02.03, группы П-403, П-404.

Теоретическая часть.

1. Понятие «инструментальное средство разработки программного обеспечения». Типы таких инструментальных средств.
2. Трансляторы. Классификация. Назначение. Примеры.
3. Возможности транслятора gcc.
4. Интегрированные среды разработки. Классификации. Назначение. Примеры. Особенности шести из следующих инструментальных средств: WEB Storm, JDeveloper, NetBeans, IntelliJ IDEA, Eclipse, KDevelop, MonoDevelop, QT Creator, XCode, PyCharm, Rational Application Developer, Android Studio.
5. Инструменты автоматизации сборки. Назначение. Примеры. Особенности шести из следующих инструментальных средств: Apache ANT, CMake, Jenkins, MSBuild, Nant, distcc, cabal, automake, autotools, scons, Apache Maven, Waf, Rake.
6. Инструмент автоматизации сборки make.
7. Инструменты автоматизации сборки automake/autoconf.
8. Назначение и использование инструментов учета обращений (issues).
9. Методики создания статической и динамической библиотек в среде POSIX.
10. Инструмент анализа покрытия кода gcov
11. Средства контроля версий. Классификация. Назначение. Примеры. Особенности шести из следующих инструментальных средств: ClearCase, CVS, Darcs, Revision Control System, Subversion, Visual Source Safe, PVCS, BitKeeper, Gnu Bazaar, Team Foundation Server, Rational Team Concert, Mercurial, Code co-op.
12. Средство контроля версий git.
13. Средства измерения производительности. Назначение. Примеры. Особенности шести из следующих инструментальных средств: Valgrind, Alinea MAP, DevPartner, LTTng, OProfile, VTune, NetBeans Profiler, CodeAnalyst, Firebug, AQtime, Intel Parallel Studio, Visual Studio Team System Profiler, средства, встроенные в Android Studio.

14. Инструмент измерения производительности и поиска утечек памяти valgrind.
15. Инструмент автоматизации тестирования AutoIT.
16. UNIT-тестирование. Инструментальные средства для UNIT-тестирования. Пример.
17. Статический анализ кода. Инструментальные средства статического анализа кода. Примеры. Особенности шести из следующих инструментальных средств: cppcheck, dafny, coverity, kiuwan, LDRA, Malpas, Polyspace, Klockwork, SemmleCode, продукты AdaCore, VeraCode, CodeRush Classic, CODAN в Eclipse, PVS-Studio, Goanna, продукты PRQA, средства IntelliJ IDEA, Astrée
18. Инструментальные средства управления требованиями (на примере, выбранном студентом).
19. DOCBOOK. Краткое описание способа использования. Достоинства и недостатки.
20. Технология wiki.
21. Особенности шести из следующих средств/способов подготовки документации: texinfo, docutils, Dr. Explain, wiki, StepShot, Help&Manual, Makrdown, AuthorIT, ClickHelp, FrameMaker, MadCap Flare, Doc-to-Help, Help Generator, StepShot, HelpStudio, FastHelp, Doc-O-Magic, Helpinator, HelpSmith, Softany Software, Daux.io, Latex
22. Инструментальные средства отслеживания ошибок. Назначение. Возможности. Примеры.
23. Инструментальные средства проектирования. Назначение. Возможности. Примеры. Особенности шести из следующих инструментальных средств: Rational Rhapsody, Astah, Papyrus, StarUML, Software Ideas Modeler, UML Designer, Enterprise Architect, Edraw Max, Modelio, Glify, yEd, BOUML, MagicDraw, PlantUML, LucidChart, Real Time Developer Studio, Umbrello, UMLet, Prosa UML Modeller, Visual Paradigm for UML, Rational Software Architect, UModel, CaseComplete, Rational System Architect, NetBeans, Microsoft Visio.
24. Средства обратной разработки. Назначение. Возможности. Пример.
25. Примеры средств автоматического и автоматизированного программирования.
26. Средство автоматизированного программирования LEX. Назначение. Возможности.
27. Инструментальные средства, используемые на базе практики
28. Выбор инструментальных средств в проекте (на примере)

Практическая часть.

Во всех заданиях практической части предполагается, что вы создаете репозиторий `git` и заносите в него все исходные файлы, разрабатываемые в рамках выполнения задания. Для программ, написанных на языке C или C++ осуществите статический анализ кода с использованием `cppcheck`.

1. Осуществите исследование влияния опций компиляции транслятора `gcc` на программу, осуществляющую вычисление суммы ряда:

$$\frac{f_1 2^1 \sqrt{1}}{1!} + \frac{f_2 2^2 \sqrt{2}}{2!} + \dots$$

Здесь $f_1 = f_2 = 1$, а последовательность f_i в целом – последовательность Фибоначчи. Суммирование следует прекращать тогда, когда слагаемое становится меньше $\epsilon = 0.000000001$

Программу следует написать оптимально.

2. Напишите программу, осуществляющую работу с файловой базой данных на языке C или C++, дающую возможность добавлять и удалять информацию об учащих (фамилия, имя, пол, оценка). Обратите внимание на проверку успешности всех выполняемых операций. Интерфейс, собственно добавление и собственно удаление разместите в различных исходных файлах. Трансляцию организуйте при помощи транслятора gcc. Сборку организуйте с помощью средства make (сборка должна быть максимально экономной: заново компилироваться должны только измененные файлы).
3. Напишите программу, осуществляющую вычисление по формуле: $(x+3*y+z\%5)^{127}$ (выражение написано с использованием синтаксиса и семантики языка C). Ввод и вывод осуществите на языке C или C++, вычисление на языке Ассемблера (исходный код, написанный на языке Ассемблера, должен размещаться в отдельном исходном файле). Сборку организуйте с помощью средства make (сборка должна быть максимально экономной: заново компилироваться должны только измененные файлы).
4. Осуществите профилирование времени выполнения программы, осуществляющей вычисление суммы ряда:

$$\frac{f_1 3^1 \sqrt{2}}{2!} + \frac{f_2 3^2 \sqrt{3}}{3!} + \dots$$

с использованием программы `valgrind`. Саму программу напишите на языке C или C++ и откомпилируйте с использованием транслятора gcc.

Здесь $f_1 = f_2 = 1$, а последовательность f_i в целом – последовательность Фибоначчи. Суммирование следует прекращать тогда, когда слагаемое становится меньше $\epsilon = 0.000000001$

Программу следует написать оптимально.

5. Напишите программу нахождения суммы всех элементов списка, удаления всех элементов, равных 0, и добавления квадрата элемента после каждого неудаленного элемента. Программа должна состоять из трех непересекающихся частей: ввод, обработка списка (обработка списка должна осуществляться за один проход списка), вывод результата. Программу напишите на языке C (не C++), трансляцию осуществите с помощью транслятора gcc, произведите исследование отсутствия утечек памяти при помощи `valgrind`.
6. На любом языке напишите программу суммирования двух чисел, использующую GUI. Проведите тестирование этой программы с использованием AutoIT.
7. Напишите программу, осуществляющую работу с файловой базой данных на языке C или C++, дающую возможность добавлять и удалять информацию об учащих (фамилия, имя, пол, оценка). Обратите внимание на проверку успешности всех выполняемых операций. Интерфейс, собственно добавление и собственно удаление разместите в различных исходных файлах. Трансляцию организуйте при помощи транслятора gcc.
Программу устройте таким образом, чтобы некоторые функции можно было протестировать автоматически, осуществите автоматическое тестирование этих функций с использованием `check`.
8. Разработайте систему требований к программе-калькулятору, имеющему интерфейс и возможности аналогичные калькулятору операционной системы Windows. Требования оформите в специализированном инструментальном средстве.
9. С использованием DocBOOK разработайте документацию для какого-либо инструментального средства, изученного в процессе прохождения дисциплины. Осуществите ее перевод в `html` или `pdf`.

10. Осуществите проектирование следующей программы с использованием всех принципов ООП, осуществите автоматическую генерацию исходных текстов на языке C++, дополните эти тексты до полностью работающего приложения.

Введите с клавиатуры товары Интернет-магазина (книги и диски). Все товары определяются ценой, книги имеют название, автора и количество страниц; диски – название и количество треков. Выведите на экран товары со стоимостью меньше 100 рублей (в том же порядке как осуществлялся ввод). При работе проверяйте корректность выполнения всех операций, количество товаров не ограничено (необходимо использовать динамические структуры данных).

11. Осуществите распознавание формулы, которая вычисляется программой, исполняемый файл которой будет вам предоставлен во время зачета.

12. С помощью flex осуществите проверку корректности арифметического выражения, написанного на языке Паскаль без использования скобок. В выражении допустимы литералы целого и вещественного типа, операции сложения, вычитания, умножения и деления.

Преподаватель _____ Глускер А. И.