

ЧАСТЬ 2

**ХІІІ МЕЖДУНАРОДНАЯ НАУЧНО-
ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ
СТУДЕНТОВ И ШКОЛЬНИКОВ
«МОЛОДЕЖЬ И ИННОВАТИКА»**

г.о.Серпухов
2020г.

@Коллектив авторов

71.	Автор: Савиц Виктор Анатольевич, студент 4 курса МАИ. Научный руководитель: Илющенко Валерий Васильевич зав. кафедрой «Информатика и информационные технологии» МАИ, к.т.н., доцент.	ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ УРОВНЕВОГО КОНТРОЛЯ СПИСКОВ САНКЦИОНИРОВАННЫХ СОБЫТИЙ	254
72.	Автор: Свинин Владислав Алексеевич студент 4 курса УЦ «Интеграция» МАИ Научный руководитель: ктн, доцент Мартынов Андрей Анатольевич	ИССЛЕДОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ИНЦИДЕНТАМИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В КОММЕРЧЕСКОЙ ОРГАНИЗАЦИИ	256
73.	Автор: Полукаров Андрей Андреевич студент 4 курса УЦ «Интеграция» МАИ Научный руководитель: доцент ИИФ Губарев Олег Константинович	АНАЛИЗ МЕТОДОВ ТЕХНИЧЕСКОЙ РАЗВЕДКИ И ПРОМЫШЛЕННОГО ШПИОНАЖА	260
74.	Автор: Прошина Анастасия Олеговна, студентка 3 курса магистратуры МАИ учебного центра «Интеграция» г. Серпухова Московской области Научный руководитель: инженер-программист Суханов Семен Валерьевич	ИДЕНТИФИКАЦИЯ ЗЛОУМЫШЛЕННИКА В ЛОКАЛЬНОЙ СЕТИ НА БАЗЕ АНАЛИЗА ЕГО АКТИВНЫХ ДЕЙСТВИЙ В ОС LINUX	262
75.	Автор: Ярошенко Вадим Васильевич, студент 4 курса МАИ. Научный руководитель: Илющенко Валерий Васильевич зав. кафедрой «Информатика и информационные технологии» МАИ, к.т.н., доцент.	ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ДАННЫХ ПУТЕМ ОБРАБОТКИ ЗАПРОСА С ИСПОЛЬЗОВАНИЕ ДИСПЕТЧЕРА ДОСТУПА К СЕТЕВЫМ РЕСУРСАМ И ОБРАБОТКИ ЗАПРОСА ДОСТУПА К ОБЪЕКТУ.	265
20. ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА			267
76.	Автор: Артемьева Анастасия, студентка 4 курса Университета «Дубна» филиала «Протвино» г. Протвино Московской области Научный руководитель: к. ф-м н., доцент Масликов Александр Альбертович	ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ КЛАССИЧЕСКИХ КРИЕТРИЕВ, ИСПОЛЬЗУЕМЫХ ПРИ МНОЖЕСТВЕННЫХ СРАВНЕНИЯХ, НА НЕГАУССОВЫХ НЕГОМОГЕННЫХ РАСПРЕДЛЕНИЯХ, МЕТОДОМ МОНТЕ-КАРЛО	267

Для всех IP—адресов в блоке 4.10 хранится ключ шифрования. Он может быть один на подсеть, выделяя эту подсеть по шифрованию данных на едином сетевом пространстве. IP—адрес сообщения (выдаваемого или принимаемого) поступает в блок 4.10 из блока 4.4, где по этому адресу выбирается ключ шифрования. Затем этот ключ передается либо в блок 4.9 при выдаче сообщения, либо в блок 4.3 при приеме сообщения. Соответственно, в блок 4.12 поступает сообщение, зашифрованное данным ключом, либо в блок 4.11 сообщение, расшифрованное выбранным по IP—адресу ключом.

Всеми действиями по осуществлению доступа управляет блок 4.5. В общем случае в блоке 4.10 хранится и собственно IP—адрес компьютера, на котором установлена система. Это необходимо для того, чтобы осуществить возможность присутствия одного компьютера сразу в нескольких подсетях, для которых ключи шифрования различны. Для однозначного выбора ключа шифрования в этом случае необходимо знание двух параметров — собственного IP —адреса и IP —адреса, с которым осуществляется взаимодействие.

20. ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

83.03.29

ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ КЛАССИЧЕСКИХ КРИТЕРИЕВ, ИСПОЛЬЗУЕМЫХ ПРИ МНОЖЕСТВЕННЫХ СРАВНЕНИЯХ, НА НЕГАУССОВЫХ НЕГОМОГЕННЫХ РАСПРЕДЕЛЕНИЯХ, МЕТОДОМ МОНТЕ-КАРЛО

Автор: Артемьева Анастасия, студентка 4 курса Университета «Дубна» филиала «Протвино» г. Протвино Московской области

Научный руководитель: к. ф-м н., доцент Масликов Александр Альбертович

Аннотация.

Методом Монте-Карло симулируется использование тестов выполняемых при множественных сравнениях на выборках малого объёма из искаженных (по сравнению с нормальным) распределений. Вычисляются эффективные вероятности ошибок 1-го рода и делаются выводы о влиянии негомогенности дисперсий, «ненормальности» эксцесса и асимметрии на эффективность исследуемых критериев.

Annotation.

The Monte Carlo method simulates the use of tests performed in multiple comparisons on small-volume samples from distorted (compared to normal) distributions. The effective probabilities of errors of the 1st kind are calculated and conclusions are drawn about the influence of inhomogeneity of variances, " abnormality " of excesses and asymmetry on the effectiveness of the studied criteria.

Ключевые слова: Математическая статистика, множественные сравнения, ANOVA.

Keywords: Mathematical statistics, multiple comparisons, ANOVA.

Отличительной чертой современной фармакологии является использование продвинутых статистических методов анализа данных. После проведения биохимического анализа формируется массив данных, который требует обработки. Выбор параметров описательной статистики основывается на типе данных и форме их распределения. Характеристику типа распределения данных дают с использованием аналитических статистических критериев.

Мы проанализируем возможности использования классических критериев Шапиро-Уилка, Левене, дисперсионного анализа (ANOVA), а также непараметрического критерия

Краскела—Уоллиса для выборок с ненормальным распределением. Нас будет интересовать – как ведут себя эти критерии на ненормальных и/или неоднородных по дисперсии выборках малого объёма.

Для исследования мы использовали программы, написанные в пакете WM. Идея в том, чтобы генерировать выборки из заданных распределений и проверять насколько частоты попадания результатов в критическую область обсуждаемого критерия близки к теоретически ожидаемым частотам.

Пакет аналитических вычислений WolframMathematica (WM) позволяет формировать компактные программы для генерации огромного числа случайных выборок (мы использовали 200000-1000000) из большого набора библиотечных распределений. Далее эти выборки можно прогонять через интересующие тесты и методом Монте-Карло получать статистически достоверные результаты. В свете формата фармакокинетических исследований нас будет интересовать случай малых выборок объёмом 5—10 элементов. Отклонения исследуемых распределений от нормального Гауссового распределения можно описывать величинами эксцесс (E) и асимметрия (A). Для нормального распределения коэффициент острровершинности $E = 3$, а $A = 0$.

Для начала, проверим корректность работы программ путём применения критериев к выборкам из нормальных распределений. Так Shapiro – Wilk, ANOVA, Kraskal-Wallis выдают значения для вероятности ошибки 1-го рода при стандартном квантиле уровня значимости $\alpha = 0.05$: 0.050, 0.051 и 0.055, что соответствует ожидаемым из теории значениям, следовательно программа работает правильно.

В классической формулировке тест ANOVA (множественные сравнения выборок на основе дисперсионного анализа) требует нормальности и однородности по дисперсиям выборок. Представляется интересным исследовать насколько катастрофичным для ANOVA будет утрата выборок нормальности и/или однородности. Рассмотрим случай 4-х выборок объёма 5 с кардинально нарушенной гомогенностью. Будем рассматривать выборки из нормальных распределений с геометрически растущим среднеквадратическим отклонением (СКО): 6, 26, 46, 86 и одинаковым мат-ожиданием (МО). Метод Монте-Карло, состоящий в данном случае в применении теста Левене к 100000 четвёрок таких выборок позволяет сделать вывод, что ~73% этих четвёрок не проходят тест на гомогенность на уровне значимости $\alpha = 0.05$. Теперь на этом же материале проведём дисперсионный анализ в стандартной форме с $\alpha = 0.05$. В результате получаем, что неоднородность приводит к эффективному увеличению вероятности ошибки 1-го рода всего до 0.10 против 0.05 в случае гомогенности дисперсий.

Далее проведём исследования для критериев Шапиро-Вилка, ANOVA, Краскела — Уоллиса по выборкам с отклонениями по эксцессу и асимметрии по сравнению с нормальным распределением. Для этого в качестве тестовых используем распределения Стьюдента, логнормальное и хи-квадрат.

t-распределение Стьюдента - это непрерывное одномерное распределение с одним параметром - количеством степеней свободы.

Пусть Y_0, Y_1, \dots, Y_n — независимые стандартные нормальные случайные величины, такие что $Y_i \sim N(0, 1)$, $i = 1, \dots, n$. Тогда распределение случайной величины t , где

$$t = \frac{Y_0}{\sqrt{\frac{1}{n} \sum_{i=1}^n Y_i^2}}$$

называется распределением Стьюдента с n степенями свободы ($df=n$).

Плотность вероятности распределения $f(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\Gamma\left(\frac{n}{2}\right)\left(1+\frac{x^2}{n}\right)^{\frac{n+1}{2}}}$.

Мы используем t-распределение с $df = 5$, которому соответствует эксцесс $E = 9$.

Логнормальное распределение — это двухпараметрическое семейство абсолютно непрерывных распределений. Если случайная величина имеет логнормальное распределение, то её логарифм имеет нормальное распределение.

Распределение случайной величины X задаётся плотность вероятности

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$$

Где $x > 0$, $\sigma > 0$, $\mu \in \mathbb{R}$. Тогда говорят, что X имеет логнормальное распределение с параметрами μ и σ .

Мы выберем параметры $\mu=0$, $\sigma=1$, тогда $E = 114$, асимметрия $A = 6.18$, мат-ожидание $M=\sqrt{e}$, медиана $m=1$.

Распределение χ^2 (хи-квадрат)-это распределение суммы квадратов n независимых стандартных нормальных случайных величин.

$$Y = X_1^2 + \dots + X_n^2, \text{ с } k=n \text{ степенями свободы.}$$

Плотность вероятности распределения

$$f(\chi^2) = \begin{cases} \left[2^{\frac{v}{2}}\Gamma(v/2)\right]^{-1} e^{-\frac{x}{2}}x^{\frac{v}{2}-1} \text{ при } x > 0, \\ 0 \text{ при } x \leq 0, \end{cases}$$

Где Γ - гамма функция, x – значение СВ χ^2 . Мы используем χ^2 -распределение с $df = 4$, которому соответствует эксцесс $E = 6$, $A = \sqrt{2}$, $M = 4$, $m = 3.35$.

Множественные сравнения (ANOVA, Краскел — Уоллис) для выборок из асимметричных распределений (логнормальное, хи-квадрат) мы исследуем в 2-х вариантах: 1) все 4 выборки асимметричны в одну сторону, 2) асимметрия 2-х выборок противоположна 2-м другим (в этом случае 2 выборки берутся отражёнными относительно M (или m) чтобы мат-ожидания (или медианы для непараметрического критерия) оставались неизменными. Итак, мы прогоняем через указанные тесты большое число (>100000) четвёрок выборок из известных распределений (используя стандартные квантили для $\alpha=0.05$) и, в соответствии с методом Монте-Карло, получаем эмпирическую вероятность ошибки 1-го рода. Результаты представлены в Таблице 1.

Таблица 1. Наблюдаемая вероятность ошибки 1-го рода (α -уровень значимости).

Выборки\Критерии	Shapiro – Wilk	ANOVA	Kruskal – Wallis
StudentTDistribution ($df = 5$)	0.066	0.045	0.056
LogNormalDistribution 4 одинаковых логнормальных распределения	0.24	0.033	0.055

LogNormalDistribution 2 логнормальных распределения 2 отражённых логнормальных	0.24	0.100	0.09
ChiSquareDistribution ($df = 4$) 4 одинаковых хи – квадрат распределения	0.098	0.044	0.055
ChiSquareDistribution ($df = 4$) 2 хи – квадратура спределения и 2 отражённых хи – квадрат	0.098	0.059	0.069

Итак, мы может сделать следующие выводы:1) негомогенность дисперсий (даже сильная) не оказывает катастрофического влияния на критерий ANOVA. В нашем примере с геометрическим ростом СКО наблюдалось возрастание эффективной вероятности ошибки 1-го рода всего с 5% до 10%. 2) Тесты множественных сравнений (ANOVA, Краскел — Уоллис) практически нечувствительны к увеличению эксцесса по сравнению с нормальным распределением при условии со направленности асимметрий. 3) Наблюдается некоторое ухудшение ситуации в случае с распределениями, имеющими разнонаправленные асимметрии.

Представляется полезным итерационным методом найти уровни значимости, соответствующие статистически наблюдаемым ошибкам 1-го рода для используемых «ненормальных» распределений. Имеет смысл провести аналогичные исследования для распределений с другими значениями параметров и другими малыми объёмами выборок. Также мы планируем аналогичным методом исследовать влияние «ненормальности» на мощность критериев.

Список использованных источников

1. Гмурман В.Е. Теория вероятностей и математическая статистика – М., Высшая школа, 2003.
2. Эверитт, Б.С. Большой словарь по статистике / Б.С. Эверитт. - М.: Проспект, 2012. - 736 с.
3. Upton, GrahamCook, Ian. Dictionaryofstatistics. -OxfordUniversityPress, Оксфорд, Великобритания, 2006. – 464 с.
4. Лабач С.Н., Чубенко А.В., Баблич П.Н., Статистические методы в медико-биологических исследованиях с использованием Excel. – 2-е изд. перераб. и доп. – К.:МОРИОН, 2001. – 408 с.
5. С. Гланц. Медико-биологическая статистика. Пер. с англ. — М., Практика, 1998. — 459 с.
6. Кобзарь А. И. «Прикладная математическая статистика», для инженеров и научных работников, 2008. Размер 8,1 Мб, 816 с.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОПЕРАЦИОННЫХ СИСТЕМ НА ПРИМЕРЕ WINDOWS, LINUX и MAC OS.

Автор: Аладина Елизавета Алексеевна, Алланурова Медине Мурадовна, студенты 3 курса «Московского Авиационного Института» филиала «Стрела» г. Серпухов Московской области.

Научный руководитель: к. п. н. доцент Михайлов Юрий Федорович, преподаватель по дисциплине «Объектно-ориентированное программирование».

Аннотация.

Данная статья посвящена проблеме выбора пользователей операционных систем. Проведен сравнительный анализ наиболее популярных ОС.

Annotation.

This article is devoted to the problem of choosing users of operating systems. For comparison, the most popular operating systems are selected. A comparative analysis of the selected operating systems.

Ключевые слова: Windows, Linux, Mac OS, ОС (операционная система), Android.

Keywords: Windows, Linux, Mac OS, OS (operating system), Android.

1. Теоретическая часть.

Особое место занимают операционные системы среди всех компьютерных программ, с которыми работают пользователи ПК.

Операционная система – комплекс управляющих и обрабатывающих программ, которые выступают как интерфейс между устройствами вычислительной системы и прикладных программ, а также предназначены для управления устройствами, вычислительными процессами. Операционная система – это мост между человеком и персональным компьютером.

От выбора ОС зависит производительность Вашей работы, степень защиты данных, необходимые аппаратные средства и т.д. Однако выбор операционной системы также зависит от технических характеристик (конфигурации) компьютера (тактовая частота процессора, оперативная и дисковая память, наличие и разрядность дополнительных карт и устройств).

Популярными среди ОС считаются WINDOWS, LINUX и MAC OS [1].

2. Основная часть.

Проведем анализ этих систем.

1. Windows:

Самая распространенная операционная система. Она установлена на 87% устройств: ноутбуки, планшеты, компьютеры. Используется ОС повсеместно. С распространением мобильных устройств – смартфонов, Linux начала теснить Windows, ведь именно она стала основой для Android. Самые главные плюсы – отличная совместимость и распространенность.

2. Linux:

LINUX менее популярен у владельцев домашних персональных компьютеров, зато более предпочтительнее на рынке смартфонов (операционная система Android имеет в основном LINUX ядро) и интернет-серверов. Более 80% серверов в Интернете работают на базе одного из дистрибутивов Linux, FreeBSD или другой Unix-подобной системы. Достоинство – оперативная доработка недочетов и неточностей благодаря открытому исходному коду.

3. MacOS:

Система, которая была разработана компанией Apple. Это сопутствующее ПО для выпускаемых этой корпорацией устройств. Основана на FreeBSD, исходный код закрыт. В

настоящий момент занимает менее 20% рынка и считается второй по популярности. Начиная с версии 10.6, система поддерживает только процессоры марки Intel. Данную систему не имеют права использовать другие производители.

Преимущества – стабильность и производительность.

График пользования ОС:

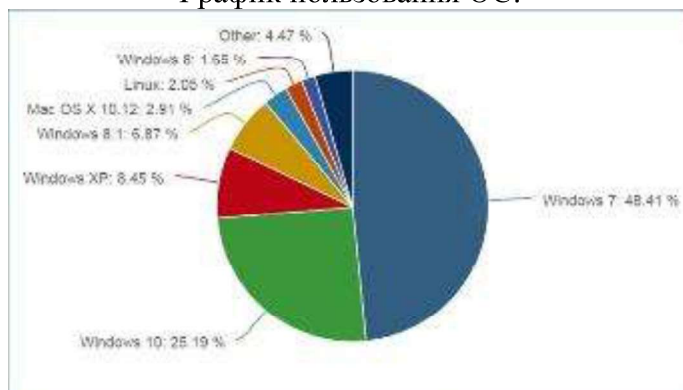


Рисунок 1

Критерии качества ОС.

Рассмотрим каждый из критериев отдельно и выясним, как "показывают" себя ОС по данным критериям.

1) Системные требования.

Конечно, на данный параметр сейчас обращают гораздо меньшее внимание, чем каких-нибудь 7–8 лет назад. Однако количество приложений, которым для работы необходимы серьезные ресурсы, увеличивается. Это значит, что и свободное место на вашем ПК или другом устройстве будет не лишним.

а) Windows.

Последние версии все более требовательны к оснащению ПК, в отличие от прошлых, "золотых", по мнению рядовых пользователей, версий - таких, как XP. Для стабильной работы системы последних версий вам понадобится процессор с двумя ядрами, 1 Гб оперативной памяти (а если берете 64-битный дистрибутив, то еще больше), не самая плохая видеокарта.

б) Linux.

Здесь ситуация проще. Достаточно одноядерного процессора, 256 Мб оперативной памяти (сразу приготовьтесь докупить "планку") и абсолютно любой видеокарты. Естественно, этого будет недостаточно для быстрой работы приложений и легкого серфинга в Интернете. Но минимальные требования именно такие. Отметим, что плюсом будет модуль ядра Zram, который позволяет сжимать данные с помощью zip перед сохранением в оперативной памяти.

в) Mac OS.

Так как система закрыта, то однозначного вывода сделать не получится. Теоретически данную ОС получится запустить с 512 Мб оперативной памяти, одноядерным процессором с частотой 1 ГГц и 9 свободными Гб памяти на жестком диске.

2) Безопасность/защищенность от вирусов.

Большинство пользователей хранят на своих компьютерах личную информацию, фото, осуществляют денежные переводы, общаются и т. д. Вся эта информация требует защиты.

а) Windows.

Считается, что это ОС наиболее уязвимая. Это можно объяснить очень просто: над последними версиями работают менее квалифицированные сотрудники. Подобное подтверждается многочисленными ошибками в коде. Если вспомнить Windows NT и

Windows XP, то процесс разработки был организован хорошо, отсюда и стабильность. Именно поэтому хакеры придумывают все новые и новые вирусы под эту оболочку. Также специалисты из Microsoft устраняют крайне мало уязвимостей, а если делают это, то через месяц или больше. Любой владелец ПК с установленной операционной системой WINDOWS должен установить надежную антивирусную программу.

б) LINUX.

Если посмотреть на LINUX, то «дыры» латаются за считанные часы. Все продукты семейства Unix имеют очень мало изъянов. Есть возможность шифрования данных, но чтобы это выполнять, потребуются определенные навыки. Что касается всплывающих блокираторов – про них можно забыть. В целом, по степени защищенности, система может бороться с MacOS.

в) Mac OS.

Наиболее безопасная ОС. Помогает поддерживать систему в стабильности за счет шифрования данных и распределения их на личные и системные. Благодаря шифрованию и четкому распределению файлов на системные и пользовательские, вирусы не могут находиться в этой среде. К тому же новая MAC OS переписана с нуля и не совместима с предыдущими версиями. Это значит, что искать пути взлома стало еще труднее.

3) Удобство в использовании.

Все разработчики стремятся сделать свои продукты максимально простыми и доступными в применении.

а) WINDOWS.

Операционные системы семейства WINDOWS известны простым и понятным интерфейсом, позволяющим сделать работу за компьютером максимально удобной. Да, он не лишен некоторых недостатков. Например, многим не по душе стандартный “Проводник WINDOWS”, поэтому они заменяют его утилитой Total Commander или похожей.

б) LINUX.

Каждый дистрибутив разрабатывается группой единомышленников из разных стран, а не отдельными людьми или фирмами. Любой может подобрать дистрибутив, исходя из рекомендаций других пользователей, с учетом своих знаний и вкусов. На данный момент, можно выделить 6 известных графических оболочек — KDE, Gnome3, Ubuntu, XFCE, Linux mint, Deepin, Rosa, Unity.

в) Mac OS.

MAC OS самая продуманная и удобная операционная система. В ней учтены все мелочи, интерфейс. Некоторые пользователи ПК устанавливают для своей ОС WINDOWS оформление в стиле операционной системы от APPLE.

3. Результаты.

Результаты анализа показывают, что применение ОС зависит от предметной области.

WINDOWS является лидером в домашнем сегменте – огромное количество поддерживаемого программного обеспечения, фирменный сервис от Microsoft и простота использования.

ОС MAC OS ориентирована на работу – стабильность и безопасность на первом месте, к тому же имеет понятный, красивый и функциональный интерфейс.

Linux системы гибкие в настройке, бесплатные и защищенные, поэтому оптимально подходят для web-разработки приложений.

4. Вывод

Таким образом, выбор должен основываться на запросах пользователей.

5. Список использованных источников

1. Самые популярные ОС среди разработчиков в России [Электронный ресурс]. – URL: <http://www.dailycomm.ru/m/43995/> (дата обращения 11.11.2019).

СОЗДАНИЕ ТЕСТОВ В СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ MOODLE

Авторы: Ананичев Кирилл Данилович, Бессонов Алексей Дмитриевич студенты 2 курса МАИ.

Научный руководитель: Старший специалист отдела СПОНГС МОУ (ИИФ) Яшин Андрей Валентинович.

Аннотация

В статье рассмотрены основные этапы создания тестов в системе дистанционного обучения. В качестве базирования были использованы тесты, которые реализуют проверку знаний по материалу. В тестах предусмотрено работа с числами.

Annotation

The article describes the main stages of creating tests in the distance learning system. It is based on tests that implement knowledge testing on the passed material. The tests provide work with numbers.

1.Цель работы:

Применяя полученные знания и основные приемы создания тестов, используя Moodle, разработать простой тест для проверки знаний.

2.Методы исследования:

1.Теоретический-используя Интернет и специальную литературу, изучили основные приемы, применяемые в ходе создания тестов.

2.Практический–применяя полученные знания, были созданы тесты.

3.Ход работы

Moodle — система, которая управляет курсами, также известная как система обучения и система проверки знаний обучающегося.

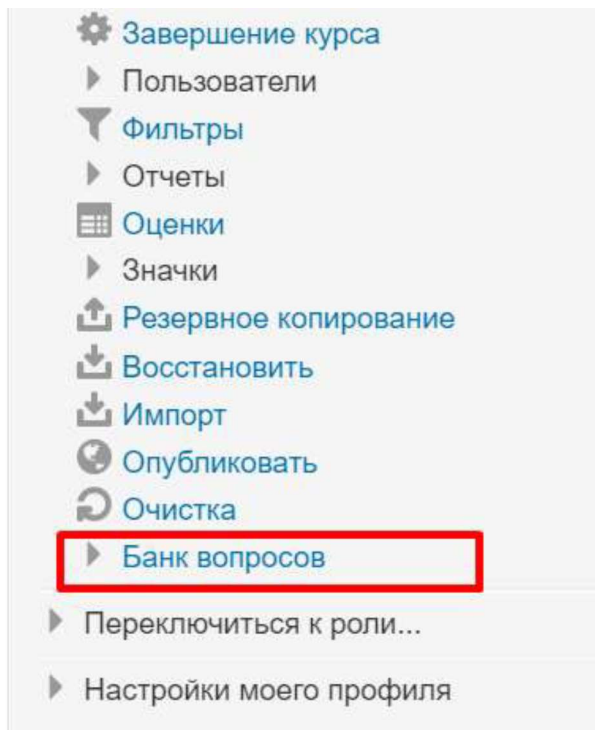
Системы дистанционного тестирования предназначена для удаленной проверки знаний учащегося. Тестируемым выдается только текст теста, хранение самих вопросов и результатов его прохождения, проверка результатов тестов, происходит на самом сервере.

Небольшое количество систем и отрыв их от Системы Дистанционного Обучения привел к тому, что системы находятся на собственных разработках, не учитывая готовые спецификации представления тестов. Существующие спецификации предполагают использование в полноценной Системе Дистанционного Обучения с курсами и участниками, а также системами анализа не только систем контроля успеваемости студента, но и ведение деятельности преподавателя. Подобные системы состоят из множества модулей, отвечающих за представление контента курса, и модуль тестирования интегрирован в другие модули. Данная модель представления курсов заставляет разработчиков систем тестирования или разрабатывать систему в уже существующей спецификации Системы Дистанционного Обучения или вовсе не использовать спецификации, так как они включают не используемые в системе тестирования составляющие Системы Дистанционного Обучения. Однако большинство малых систем также разрабатывает модули для экспорта/импорта тестов в другие системы, в том числе и для систем, использующих разные спецификации.

Рассмотрим технологии создания вопросов на примере теста перевода чисел из двоичной СС в восьмеричную. Для этого перейдем в раздел «Тестирование» и в пункте Редактировать». Для примера выбран курс «Системы счисления». Создаваемый вопрос будет помещен в банк вопросов для данного курса и будет доступен для просмотра, редактирования и добавления в тесты.

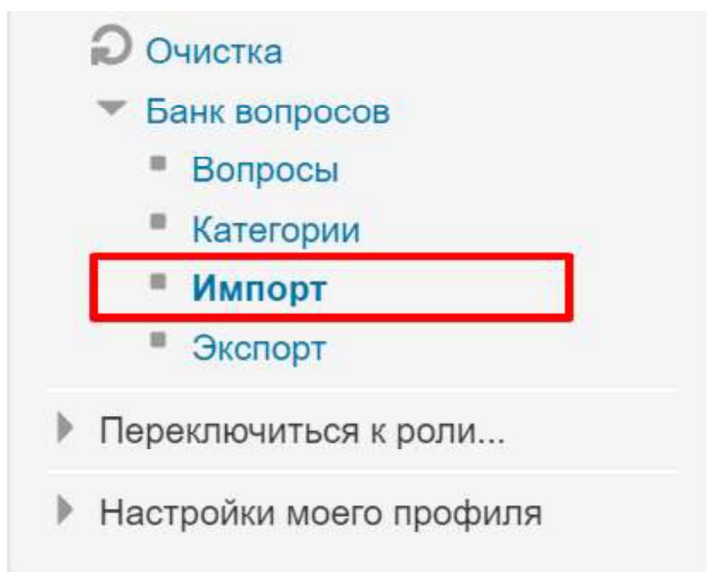
Вспользуемся технологией создания тестов на основе банка заданий (вопросов) в Moodle. Для того, чтобы добавить новый тест необходимо выполнить определенные действия.

1) В выпадающем списке ресурсов электронного курса выбрать «Банк вопросов». (см. рис.1)



(рис.1)

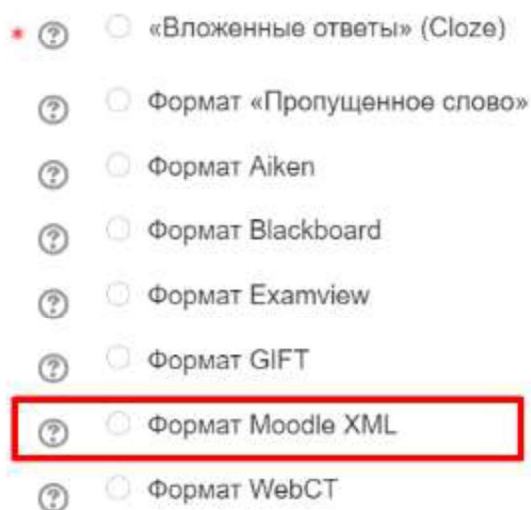
2) В выпадающем списке ресурсов электронного курса выбрать «Импорт». (см. рис.2)



(см. рис.2)

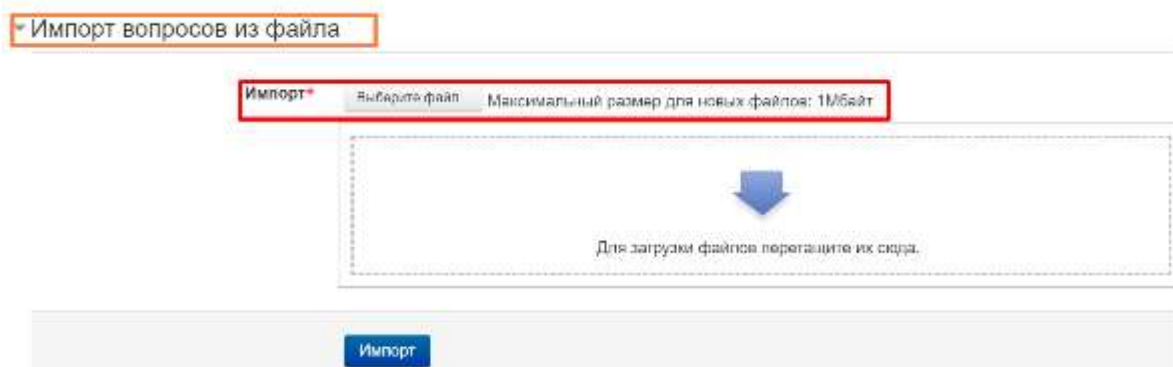
3) Открыв окно выбрать формат тестов «Формат Moodle XML». (см. рис.3)

▼ Формат файла



(см. рис.3)

4) В окне «Импорт вопросов из файла» нажмите на «Выберите файл», выбираете нужный тест и нажмите кнопку «Импорт» (см. рис.4)



(см. рис.4)

5) После всех проделанных действий тест можно пройти во вкладке «Просмотр».

6.Вывод: Теоретически и на практике мы изучили основные приемы создания тестов в СДО, используя Moodle

Список использованных источников:

1. Создание тестов в системе дистанционного обучения moodle 2.2 [Электронный] URL <https://docplayer.ru/30041469-Sozdanie-testov-v-sisteme-distancionnogo-obucheniya-moodle-2-2.html>
2. Тестирование в системе Moodle [Электронный] URL <https://xn----7sbbzid0ad2blh2czd.xn--p1ai/internet/22-moodle/14-testirovanie-v-sisteme-moodle>

3. Использование системы Moodle для создания тестов по дисциплине «Физика»[Электронный] URL <https://pandia.ru/text/80/382/4093.php>

ПОДДЕРЖКА И РАЗВИТИЕ ИГРЫ SPACE STATION 13 НА ПЛАТФОРМЕ BYOND ЯЗЫКА DREAM MAKER

Автор: Ахтырский Андрей Игоревич, студент первого курса филиала “Протвино” государственного университета “Дубна”, кафедра информационных технологий

Научный руководитель: к.т.н. Астафьева Марина Петровна, доцент кафедры информационных технологий филиала “Протвино” государственного университета “Дубна”

Аннотация

В данной работе описано принятие участия в разработке и поддержке развития серверов игры Space Station 13. Цель работы — показать приёмы разработки и поддержки игрового приложения с применением кода, расположенного в открытом доступе на Github.com. Используемые платформы и программные продукты: BYOND (Build Your Own Net Dream), Dream Maker.

Annotation

This work describes the participation in the development and supporting of progress in game servers Space Station 13. The purpose of this work is show development and supporting game application using open-based code from Github.com. Used platforms: BYOND, Dream Maker.

Компьютерные игры – важный аспект нашего общества. Они являются актуальными и по сей день, постоянно развиваясь и все время показывая что-то новое. Для людей такое увлечение играет разную роль: для кого-то это заработок в киберспорте или разработке, для кого-то – способ отдохнуть, получить наслаждение от красивой картинке или от игрового процесса. Также в настоящее время интенсивно развиваются открытые платформы и работа при помощи удаленного доступа с открытым кодом базы, что обеспечивает актуальность данной работы.

Задачи, которые необходимо решить для достижения цели работы: опрос и анализ требований пользователей, нахождение и изучение работы определенного кода программы в открытом доступе, создание графического изображения, написание программного кода, тестирование разработанного фрагмента игры.

Используемые платформы и программные продукты: BYOND (Build Your Own Net Dream) – платформа для создания различных двумерных игр, работающая под операционными системами Windows и Linux. Созданные игры на этой платформе разработаны исключительно для использования на компьютере. Данная платформа также использует для программирования свой собственный язык Dream Maker. Игровое сообщество является межнациональным. Люди разных стран играют на серверах, вместе разрабатывают новые идеи и игры. Объектом исследования являются используемые платформы и программные продукты применительно к рассматриваемой разработке.

Dream Maker основан на методе наследования – есть “родительские” объекты, которые передают свои переменные “дочерним” объектам. BYOND обладает базой объектов, которые легко позволяют создать новые объекты и передавать им различные значения. Эти объекты называются atom, и включает следующие компоненты: Area – определенная зона, размером в один тайл – мельчайшей единицей измерения пространства в BYOND; Turf – “покрытие” мира (пол, стены, возможности их редактирования); Obj – игровые объекты, которые можно двигать, заставить двигаться или просто использовать по назначению; Mob – игровой объект, который может быть подконтролен персонажу. База кода на многих

серверах открыта, что позволяет изучать её для саморазвития или её улучшения, и расположена, преимущественно, на Github.com.

Space Station 13 – многопользовательская ролевая компьютерная игра о работе и жизни персонала на космической станции. Первая версия игры была разработана и выпущена в свет в 2003 году. В игре существуют различные игровые сервера, владельцы и разработчики которых разрабатывают игру по различным направлениям. Основная черта игрового процесса – отыгрывание определенных профессий на космической станции: инженер, детектив, член службы безопасности, врач, капитан и так далее. Игра является сессионной, т.е. каждый раунд длится определенное время, пока не произойдет какое-то критическое событие в игре или пока персонал не вызовет эвакуационный шаттл. Каждый раунд происходят различные события: нападение космических рейдеров, агентов синдиката, неизвестных инопланетных существ и многое другое. Игрок управляет определенным персонажем, который может передвигаться вверх, вниз, влево и вправо, на более современных серверах – ещё и по осе Z. (между этажами станции). Управление происходит с помощью клавиатуры, мышкой. Игрок может взаимодействовать при помощи мышки с различными игровыми объектами: автоматами по продаже напитков, компьютерами, шкафчиками с различным содержимым, газами для разных целей и т.д.

На рисунке 1 можно увидеть экран игры. Правую половину экрана занимает панель с различными вкладками, отвечающими либо за кнопки быстрого взаимодействия с окружающим миром, либо за информацию о состоянии сервера, игровом мире и т.д. На левой части экрана изображена игровая область – на ней можно увидеть главного персонажа по центру и различные объекты, расположенные вокруг него. В нижней части экрана находится строка ввода, в которую можно вводить команды. К примеру: say”Привет мир выведет на экран сообщение от лица персонажа Привет мир, которое будет видно другим персонажам в радиусе видимости говорящего. Над строкой ввода располагается HUD персонажа: инвентарь, состояние персонажа, его здоровье и т.д.



Рисунок 1 – Игровая область

Далее приведены несколько примеров решения различных подзадач. На Рисунке 2 представлен пример программного кода, который обрабатывает сборку оружия по частям в игре.

Результат программного кода можно увидеть на Рисунке 3. Как можно увидеть, первый объект — приклад, второй объект — основа оружия, третий объект — основа оружия

с предустановленным прикладом, четвертый объект — дуло оружия. При нажатии приклада

```
/obj/item/weapon/weaponparts/heavysniper/stocked/attachby(obj/item/W, mob/user)
  if(istype(W,obj/item/weapon/weaponparts/heavysniper/barrel))
    to_chat(user, "You attach the barrel to the sniper")
    var/obj/item/weapon/gun/projectile/heavysniper/HS = new (get_turf(src))
    if(loc == user)
      equip_slot = user.get_inventory_slot(src)
      if(equip_slot in list(slot_r_hand, slot_l_hand))
        user.drop_from_inventory(src)
        user.equip_to_slot_if_possible(HS, equip_slot)
    qdel(W)
    qdel(src)
```

на
осн
ову
пол
уча
етс
я
тре
тий
объ
ект,
дал
ее
доб
авл

яется дуло таким же способом. Результат — оружие.

Рисунок 2 — Фрагмент программного кода присоединения к полусобранному оружию ствола



Рисунок 3 — Сборка оружия в игре

На Рисунке 4 представлен код создания коробки с запчастями для оружия и добавление её в список покупаемых предателями предметов, а также сам предмет.

```

/obj/item/weapon/storage/box/syndie_kit/antimaterial_rifle
    name = "Sniper rifle box"
    desc = "Sniper rifle kit. One shot for real men."
    icon_state = "box_of_doom_big"
    w_class = ITEM_SIZE_HUGE

/obj/item/weapon/storage/box/syndie_kit/antimaterial_rifle/New()
    ..()
    new /obj/item/ammo_casing/a145(src)
    new /obj/item/weapon/weaponparts/heavysniper/disassembled(src)
    new /obj/item/weapon/weaponparts/heavysniper/stock(src)
    new /obj/item/weapon/weaponparts/heavysniper/barrel(src)

/datum/uplink_item/item/visible_weapons/heavysniper
    name = "Anti-material Rifle"
    item_cost = 20
    path = /obj/item/weapon/storage/box/syndie_kit/antimaterial_rifle

```



Рисунок 4 — Код создания коробки под оружие и спрайт самого объекта

Оператор `name` отвечает за присваивание имени объекту. `Desc` – его описание, `icon_state` – спрайт объекта. Путь объявляется в родительском объекте оператором `icon`. `w_class` – размер объекта. Под объектом `obj/item/weapon/storage/box/syndie_kit/antimaterial_rifle/New()` расположено создание предметов из списка, приведенного под объявлением объекта. `New` – добавляет объявленный в списке объект, `(src)` – расположение нового предмета (`src` означает работу с тем объектом, под которым оно объявлено, то есть в нашем случае в саму коробку).

```

/obj/item/clothing/glasses/powered/thermal/lens
    name = "Thermal lenses"
    desc = "Lenses for glasses."
    toggleable = FALSE
    icon_state = "thermal_lens"
    body_parts_covered = 0
    slot_flags = 0
/obj/item/clothing/glasses/attackby(var/obj/item/I, var/mob/user)

    if (istype(I,/obj/item/clothing/glasses/powered/thermal/lens))
        overlay = global_hud.thermal
        vision_flags = SEE_MOBS
        see_invisible = SEE_INVISIBLE_MOLIGHTING
        protection = flash_protection
        flash_protection = FLASH_PROTECTION_REDUCED
        origin_tech = list(TECH_ILLEGAL = 3)
        to_chat(user, "You attached your lenses to your glasses")
        have_lenses = 1
        qdel(I)

```

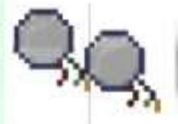


Рисунок 5 — Код создания специальных линз для очков, которые позволят видеть живых существ сквозь стены

В данном случае мы также, как и со сборкой оружия, сначала создаем объект, присваивая ему различные переменные, объявленные в родительских объектах (`toggleable` – требуется ли включать объект, `body_parts_covered` – какие части тела он покрывает на спрайте, `slot_flags` – в какие части одежды можно вставлять этот предмет. Последним двум значениям присваивается 0, так как этот объект – модификация другого, очков, и некоторые значения нам требуется изменить.) Мы также применяем `proc`. Сначала прописываем объект