

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЁТ
по ознакомительной практике

Выполнил:

Д. С. Семёнов

Студент группы
321703

Проверил:

В. Н. Тищенко

Минск 2024

СОДЕРЖАНИЕ

Введение	3
1 Постановка задачи	4
2 Формализованные фрагменты теории интеллектуальных компьютер- ных систем и технологий их разработки	5
3 Формальная семантическая спецификация библиографических ис- точников	12
Заключение	16
Список использованных источников	17

ВВЕДЕНИЕ

Цель:

Закрепить практические навыки формализации информации в интеллектуальных системах с использованием семантических сетей.

Задачи:

- Построение формализованных фрагментов теории интеллектуальных компьютерных систем и технологий их разработки;
- Построение формальной семантической спецификации библиографических источников, соответствующих указанным выше фрагментам;
- Оформление конкретных предложений по развитию текущей версии Стандарта интеллектуальных компьютерных систем и технологий их разработки

1 ПОСТАНОВКА ЗАДАЧИ

Часть 2 Учебной дисциплины "Представление и обработка информации в интеллектуальных системах"

⇒ библиографическая ссылка*:

- монография *OSTIS*
- Материалы конференций *OSTIS*
- М.П. Малыхина, Д.А. Герасимов..МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА
⇒ URL*:
[<https://ntk.kubstu.ru/data/mc/0051/2074.pdf>]
- Определение и проблемы языков программирования
⇒ URL*:
[<https://ppt-online.org/149158>]
- Корниевич А.И, Гардейчик С.М..МЕТАПРОГРАММИРОВАНИЕ КАК ОДИН ИЗ ВИДОВ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ
⇒ URL*:
[<https://rep.bntu.by/bitstream/handle/data/106728/7-8.pdf?sequence=1>]
- Кроссплатформенность
⇒ URL*:
[<https://secretmag.ru/enciklopediya/chto-takoe-krossplatformennost-obyasnyаем-prostymi-slovami.htm>]
- Шарикова М.И..Определение и проблемы языков программирования
⇒ URL*:
[<https://novainfo.ru/article/8173>]
- Ясницкий Л.Н..Интеллектуальные системы
⇒ URL*:
[<https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf>]
- Степанов О.Г..Автоматное программирование
⇒ URL*:
[<https://cyberleninka.ru/article/n/avtomatnoe-programmirovaniye-s-ispolzovaniem-dinamicheskikh-yazykov-programmirovaniya>]

2 ФОРМАЛИЗОВАННЫЕ ФРАГМЕНТЫ ТЕОРИИ ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ И ТЕХНОЛОГИЙ ИХ РАЗРАБОТКИ

мультиагентная система

:= [направление искусственного интеллекта, которое использует систему агентов для решения сверхсложных задач или глобальных проблем]

⊂ *искусственный интеллект*

⇒ *примечание**:

[агенты в мультиагентных системах взаимосвязаны между собой и обладают возможностью обмена сообщениями. У каждого агента есть своя цель и решаемая им задача. Обычно в мультиагентных системах используются программные агенты. Тем не менее, составляющими мультиагентных систем могут быть также роботы, люди или команды людей]

⇒ *основные требования при формировании мультиагентных систем**:

{• *автономность*

⇒ *пояснение**:

[Все агенты должны быть независимы в управлении, а поведение агентов прикладных систем определяется их ролью в этой системе. При этом агенты могут иметь свои индивидуальные цели, которые отличаются от целей системы. В соответствии с этим они способны самостоятельно планировать собственное поведение для достижения своих целей, осуществляя самоконтроль над своими действиями и внутренним состоянием, а также контролировать узлы связи агентов.]

• *децентрализация*

⇒ *пояснение**:

[Агенты наделяются правами и полномочиями для реализации автономного поведения и способности принимать решения. Таким образом, агенты могут выполнять не только команды извне, но и обмениваться сообщениями, на основе которых принимают решения. Иногда в виде исключения применяются посредники в виде агентов, ответственных за реализацию протокола передачи сообщений между агентами.]

• *взаимодействия*

⇒ *пояснение**:

[Автономность и децентрализация не исключают, а наоборот, предполагают активные взаимодействия между агентами. Поведение агентов предполагает обмен информацией, координацию действий, согласование решений и т.п. Способность к взаимодействию агентов имеет принципиальное значение, поскольку именно от нее в большинстве случаев зависит возможность обеспечения и качество реализации всех других свойств. Это так называемые "вычисления, основанные на взаимодействиях". Таким образом, достигается максимальная фокусировка агента на своей задаче и сокращается время работы агентной системы]

• *распределенность*

⇒ *пояснение**:

[Использование мультиагентных систем является естественным подходом для разработки приложений распределенных систем, при котором происходит разброс агентов на разных вычислительных устройствах или процес-

сорах, что может существенно повышать производительность прикладных систем. Тем не менее, это вовсе не исключает преимуществ использования мультиагентного подхода для решения прикладных задач, когда агенты функционируют на одном вычислительном устройстве в параллельном режиме]

- *симметричность*

⇒ *пояснение**:

[При обмене сообщениями между агентами, каждому из них должны быть предоставлены одинаковые возможности участия в переговорах для организации корректного получения информации]

- *устойчивость*

⇒ *пояснение**:

[При обмене сообщениями между агентами каждый из них не должен позволять другим агентам извлекать дополнительную пользу при отклонении от установленных правил. Таким образом, обмен сообщениями должен обеспечить достижение состояния равноправия между агентами]

- *простота*

⇒ *пояснение**:

[Для повышения скорости обмена сообщениями и ускорения работы всей мультиагентной системы обмен сообщениями не должен предъявлять дополнительных требований к архитектуре агентов и усложнять структуру коммуникационных протоколов]

- *безопасность*

⇒ *пояснение**:

[При разработке мультиагентной системы должна обеспечиваться ее защищенность во избежание несанкционированного доступа к компьютерной информации. Несанкционированный доступ может повлечь за собой как сбой в работе мультиагентной системы, так и утрату или искажение важной информации]

- *интеллектуальность*

⇒ *пояснение**:

[Мультиагентная система должна отвечать требованиям интеллектуальности, то есть содержать в себе интеллектуальные методы. Для повышения эффективности можно применять базы данных с использованием интеллектуальных методов]

}

⇒ *автор**:

- *М.П. Малыхина*
- *Д.А. Герасимов*

⇒ *библиографическая ссылка**:

- *МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА*
- *<https://ntk.kubstu.ru/data/mc/0051/2074.pdf>*

язык программирования

:= [знаковая система для планирования поведения компьютера]

:= [правила образования знаков и согласованные с ними правила образования денотатов]

⇒ *примечание**:

[денотат - значение, смысл]

- ⇒ *примечание**:
 [знак - обозначение или имя]
- := [средство общения между человеком и компьютером]
- := [инструмент для производства программных услуг]
- ⇒ *основные предметные области знаний применения языков программирования**:
 {• *научные вычисления*
 • *обработка деловой информации*
 • *искусственный интеллект*
 • *системная область*
 • *Web-обработка*
 }
- ⇒ *основные критерии эффективности языка программирования**:
 {• *читабельность*
 ⇒ *примечание**:
 [Характеристика читабельности подразумевает, что текст программ должен быть легко понимаемым, чтобы разработчики могли быстро читать и анализировать код.]
- *легкость создания программ*
 ⇒ *примечание**:
 [Легкость создания программ предполагает наличие удобных средств и синтаксиса для создания программ, особенно в специфических областях применения.]
- *надежность*
 ⇒ *примечание**:
 [Высокая надежность требует минимизации ошибок в работе программного обеспечения, что обеспечивает стабильность и предсказуемость поведения программ.]
- *стоимость*
 ⇒ *примечание**:
 [Снижение затрат на протяжении всего жизненного цикла программ включает уменьшение расходов на разработку, тестирование, выполнение и сопровождение программного обеспечения.]
- *переносимость программ*
 ⇒ *примечание**:
 [Переносимость программ подразумевает способность программного обеспечения работать на различных платформах без значительных изменений кода.]
- *универсальность*
 ⇒ *примечание**:
 [Универсальность языка программирования позволяет применять его для решения широкого круга задач в разных областях.]
- *четкость*
 ⇒ *примечание**:
 [Четкое и полное официальное описание языка программирования обеспечивает его точное понимание и правильное использование.]
- }
- ⇒ *актуальные проблемы языков программирования**:
 {• *аппаратная сложность*
 ⇒ *примечание**:

[Аппаратная сложность опережает наше умение строить ПО, использующее потенциальные возможности аппаратуры.]

- *отставание разработки*

⇒ *примечание**:

[Умение разрабатывать новые программы отстает от требований к новым программам.]

- *качество существующих программ*

⇒ *примечание**:

[Возможности эксплуатировать существующие программы угрожает низкое качество их разработки.]

}

⇒ *библиографическая ссылка**:

- *Определение и проблемы языков программирования*
- <https://ppt-online.org/149158>

парадигма программирования

:= [исходная концептуальная схема или модель, принятая в качестве образца постановки проблем и их решения]

:= [результат анализа и обобщения способа прогрессивного научного мышления, характерного для данной области]

⇒ *примечание**:

[Впервые понятие парадигмы ввел в естествознание Т. Кун. Выбор той или иной парадигмы для решения конкретной проблемы определяет методологическое обоснование методов ее решения и задает регулятивные основы мышления разработчика.]

⇒ *разбиение**:

- { • *операционная парадигма*
- *императивная парадигма*
- *функциональная парадигма*
- *структурная парадигма*
- *объектно-ориентированная парадигма*
- *последовательная парадигма*
- *параллельная парадигма*
- *потокосная парадигма*
- *трансформационная парадигма*
- *асинхронная парадигма*
- *синхронная парадигма*
- *дефиниционная парадигма*
- *логическая парадигма*
- *демонстрационная парадигма*
- *сетевая парадигма*
- *локальная парадигма*
- *парадигма капсулизации*

}

⇒ *автор**:

- *В С. Выхованец*
- *О.Д. Выхованец*

⇒ *библиографическая ссылка**:

http://valery.vykhovanets.ru/Texts/1997/Vykhovanets1997_2.pdf

метапрограммирование

:= [вид программирования, связанный с созданием программ, порождающих другие программы как результат своей работы]

⇒ *примечание**:

[Метапрограммирование применяется для повышения функциональности программ ценой меньших затрат (объем кода, усилия на сопровождение и т.п.). Его характерной особенностью является то, что часть необходимых пользователю вычислений выполняется на этапе трансляции программы.]

⇒ *автор**:

- *Гардейчик С.М.*
- *Корниевич А.И.*

⇒ *библиографическая ссылка**:

- *МЕТАПРОГРАММИРОВАНИЕ КАК ОДИН ИЗ ВИДОВ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ*
- *<https://rep.bntu.by/bitstream/handle/data/106728/7-8.pdf?sequence=1>*

кроссплатформенность

:= [способность IT-продукта полноценно работать на любом устройстве вне зависимости от типа операционной системы]

⇒ *примечание**:

[Задача программистов, обеспечивающих кроссплатформенность — написать такой код, чтобы пользователи смогли получить доступ к сервису с устройств, работающих на любой операционной системе]

⇒ *библиографическая ссылка**:

<https://secretmag.ru/enciklopediya/chto-takoe-krossplatformennost-obyasnyаем-prostymi-slovami.htm>

семантический разрыв

:= [различие между описаниями объектов различных формальных представлений, например, языкового и символьного описаний]

:= [несоответствие выводимого результата запросу пользователя]

⇒ *примечание**:

[пользователь при формировании запроса оперирует понятиями высокого уровня, такими как имя объекта. При этом существующие системы поиска данных оперируют низкоуровневыми признаками, такими как цвет, форма и текстура]

⇒ *автор**:

Шарикова М.И.

⇒ *библиографическая ссылка**:

<https://novainfo.ru/article/8173>

экспертная система

:= [сложный программный комплекс, аккумулирующий знания специалистов в конкретной предметной области и использующий эти знания с целью выработки логически обоснованных рекомендаций и решений проблем, а также для консультаций менее квалифицированного пользователя]

⇒ *автор**:

Ясницкий Л.Н.

⇒ *библиографическая ссылка**:

- *Интеллектуальные системы*
- *<https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf>*

интерфейс разработчика

- := [программа, с помощью которой инженер-когнитолог и программист могут создавать базу знаний в диалоговом режиме. Включает в себя системы вложенных меню, шаблонов языка представления знаний, подсказок (help-режим) и других сервисных средств, облегчающих работу с базой знаний.]
- ⇒ автор*:
Ясницкий Л.Н.
- ⇒ библиографическая ссылка*:
 - *Интеллектуальные системы*
 - <https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf>

интерфейс пользователя

- := [комплекс программ, реализующих диалог пользователя с экспертной системой на стадии как ввода информации, так и получения результатов]
- ⇒ автор*:
Ясницкий Л.Н.
- ⇒ библиографическая ссылка*:
 - *Интеллектуальные системы*
 - <https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf>

SWITCH-технология

- := [автоматное программирование]
- := [программирование с явным выделением состояний]
- ⇒ примечание*:

[Одной из основных SWITCH-технологий является графический язык, позволяющий описывать поведение систем со сложным поведением в терминах состояний и переходов между ними и связи между этими системами. При использовании SWITCH-технологии в разработке программного обеспечения важной частью разработки является реализация поведения, описанного для диаграммах переходов, на целевом языке программирования.]
- ⇒ подходы для реализации поведения систем*:
 - {• *полностью ручное программирование*
- ⇒ примечание*:

[Одним из простейших общепринятых методов ручного программирования является следующий: текущее состояние системы хранится в переменной интегрального или перечислимого типа, а основная логика программы сосредоточена внутри одного или нескольких операторов *switch*, определяющих действия программы в зависимости от текущего состояния. Другим популярным методом является использование паттерна программирования State. Несмотря на такие преимущества этой группы методов, как высокая производительность и полный контроль над получаемым кодом, она имеет существенные недостатки — низкая читаемость кода и большая трудоемкость.]

 - *автоматическая генерация кода по диаграмме переходов*
- ⇒ примечание*:

[Обычно при этом подходе генерируется код, аналогичный тому, который получается при использовании ручного программирования. Недостатками этого подхода являются низкая читаемость кода (связанная с тем, что в качестве целевого языка используется императивный язык, например *Java*, и при переносе теряется информация, специфичная для логики диаграмм

переходов), малая степень контроля над получаемым кодом, невозможность ручного изменения получаемого кода и привязанность к конкретному формату входных данных, с использованием которого задается исходная диаграмма переходов]

- *Ручное написание кода с использованием специальной библиотеки*

⇒ *примечание**:

[Ручное написание кода с использованием специальной библиотеки. В этом случае происходит перенос диаграммы переходов в вызовы специальной библиотеки, которая по этим инструкциям строит внутреннее представление диаграммы переходов. Затем, согласно этому представлению, происходит исполнение логики описанного автомата. Основным преимуществом подхода является то, что вызовы библиотеки отражают семантику диаграммы переходов каждый вызов может, например, соответствовать объявлению состояния или перехода), что позволяет создавать читаемый код, который легко поддерживать. Также некоторые библиотеки ориентированы на конкретные виды взаимодействия автоматного и объектно-ориентированного кода, что позволяет более гладко комбинировать эти подходы к программированию. Основными недостатками являются низкая производительность некоторых из таких систем (существуют системы, основанные на мета-программировании и статической генерации кода, которые позволяют достичь высокой производительности), а также невозможность описать ряд конструкций диаграмм переходов, используя ограниченный синтаксис целевого языка]

}

⇒ *автор**:

Степанов О.Г.

⇒ *библиографический источник**:

<https://cyberleninka.ru/article/n/avtomatnoe-programmirovanie-s-ispolzovaniem-dinamicheskikh-yazykov-programmirovaniya>

3 ФОРМАЛЬНАЯ СЕМАНТИЧЕСКАЯ СПЕЦИФИКАЦИЯ БИБЛИОГРАФИЧЕСКИХ ИСТОЧНИКОВ

Мультиагентные системы искусственного интеллекта

⇒ *тип источника**:

[статья]

⇒ *ключевой знак**:

- *агент*
- *обмен сообщениями*
- *поведение агента*
- *принятие решений*
- *мультиагентная система*
- *искусственный интеллект*

⇒ *аннотация**:

[Проблема управления предприятием или системой в режиме реального времени - важный аспект во многих отраслях промышленности, транспорта, банковского дела и даже в космосе. На помощь в решении проблем обработки больших объемов данных, в том числе и повышении производительности этих процессов, приходят мультиагентные системы искусственного интеллекта. В данной статье проанализированы основы технологий построения прикладных систем группового управления, состоящих из большого числа автономных подсистем, организованных в сеть, узлы которой могут работать под управлением различных операционных систем и в различных коммуникационных средах. Подобную ситуацию хорошо иллюстрируют опубликованные в открытом доступе результаты оценки эффективности применения такого рода систем для автономного управления грузопотоком на международной космической станции и в крупнейшей корпоративной системе такси Лондона]

⇒ *цитата**:

[Мультиагентные системы универсальны и отлично справляются с задачами любой сложности]

⇒ *цитата**:

[В течение последнего времени произошел большой скачок информационных технологий, а также значительный рост различных транспортных сетей и узлов. Анализ транспортных сетей показывает, что мультиагентные системы способны справиться со спецификой компании и повысить эффективность ее работы от 3 до 15%, которая влечет за собой экономическую выгоду. Такого рода задачи с большими объемами факторов наиболее подходят для применения в мультиагентных системах за счет использования таких их свойств, как: автономность, взаимодействие, распределенность и т.д. Все вышесказанное позволяет сделать вывод о том, что мультиагентные системы универсальны и отлично справляются с задачами любой сложности.]

Определения и проблемы языков программирования

⇒ *тип источника**:

[презентация]

⇒ *оглавление**:

[

- анализ определения понятия "язык программирования"
- области применения языков программирования

- виды языков программирования
- развитие и стандартизация языков программирования

]

⇒ цитата*:

[Основная задача XXI века - это совершенствование качества компьютерных систем, возможности которых целиком определяются программным обеспечением]

⇒ цитата*:

[Технологический критерий качества языка программирования - язык тем лучше, чем проще производство на его основе программных услуг]

Что такое кроссплатформенность. Объясняем простыми словами

⇒ тип источника*:

[статья]

⇒ цитата*:

[Особенно актуальной кроссплатформенность стала в 2010-е годы на фоне резкого роста числа мобильных посетителей интернета.]

⇒ цитата*:

[Адаптировать код под различные ОС можно с помощью современных кроссплатформенных языков программирования (например, С, С++). При надлежащем качестве кода достаточно будет переписать только некоторые фрагменты, не трогая при этом основной движок программы.]

⇒ цитата*:

[Следует отличать понятие кроссплатформенности и кроссбраузерности. Кроссплатформенность - Кроссплатформенность — это способность IT-продукта полноценно работать на любом устройстве вне зависимости от типа операционной системы (ОС). Кроссбраузерность - способность сайта работать в Google Chrome, Opera, Mozilla Firefox и других браузерах, не теряя при этом скорости работы и читабельности страниц. Задача разработчика – сделать портал пригодным для удобного использования в любом браузере.]

⇐ сравнение*:

- Кроссплатформенность
- Кроссбраузерность

Параллельная парадигма программирования и её использование в языках высокого уровня

⇒ тип источника*:

[статья]

⇒ оглавление*:

[

- Парадигмы программирования
- Параллельное программирование
- Особенности реализации
- Применение

]

⇒ цитата*:

[Существуют задачи, эффективное решение которых сопряжено со значительными трудностями при использовании последовательной модели вычислительного процесса и которые описываются множеством параллельно существующих и взаимодействующих объектов (процессов) различной природы. Эти задачи, как правило, имеют высокую трудоемкость и сложность программирования]

Метапрограммирование как один из видов современных технологий программирования

⇒ *тип источника**:

[статья]

⇒ *цитата**:

[Метапрограммирование применяется для повышения функциональности программ ценой меньших затрат (объем кода, усилия на сопровождение и т.п.). Его характерной особенностью является то, что часть необходимых пользователю вычислений выполняется на этапе трансляции программы.]

⇒ *цитата**:

[Основной вид метапрограммирования — это метапрограммирование шаблонов (template metaprogramming, TMP). TMP реализуется с помощью шаблонов языка программирования C++. Технология TMP даёт свои преимущества. Поскольку шаблонные метапрограммы исполняются во время компиляции C++, они могут перевести часть работы со стадии исполнения на стадию компиляции. Ещё одно преимущество состоит в том, что программы C++ с использованием TMP можно сделать эффективными почти во всех смыслах: компактность, быстродействие исполняемого кода, экономия памяти.]

Иной взгляд на проблему «семантического разрыва»

⇒ *тип источника**:

[статья]

⇒ *аннотация**:

[В статье приведены рассуждения на тему термина «семантический разрыв», а также о целесообразности применения данного термина в контексте систем распознавания изображений. Сделан вывод о несоответствии смыслового содержания и определения термина. На основании рассуждений введено определение термина «семантика» применительно к системам распознавания.]

⇒ *ключевой знак**:

- *семантика*
- *система распознавания изображения*
- *семантический разрыв*

⇒ *цитата**:

[Можно трактовать понятие семантики как суть, сущность предмета, передаваемая с помощью языка. А суть по своему определению — это то, для чего предмет есть.]

⇒ *цитата**:

[Само слово «семантический» означает «подлежащий семантике», а «семантика» - содержание, информация, передаваемые на языке или какой-либо его единицей. То есть «семантический разрыв» можно охарактеризовать как несоответствие выводимого изображения запросу пользователя.]

Интеллектуальные системы

⇒ *тип источника**:

- ⇒ [книга]
оглавление*:
[
• Прошлое и настоящее искусственного интеллекта и интеллектуальных систем
• Модели представления знаний
• Экспертные системы
• Понятие о классической нейронной сети
• Возможности и сферы применения нейронных сетей
• Оптимальное проектирование и обучение нейронных сетей
• Неклассические нейронные сети
• Компьютерное творчество
• Настоящее и будущее искусственного интеллекта и интеллектуальных систем
]

Автоматное программирование с использованием динамических языков программирования

- ⇒ тип источника*:
[статья]
⇒ аннотация*:
[В работе рассматривается проблема переноса диаграмм переходов, принятых в SWITCH-технологии, в исполняемый код. Для решения данной проблемы предлагается использование динамических языков программирования, возможности которых позволяют добиться изоморфности диаграммы переходов и соответствующего программного кода. Разработан формальный метод преобразования диаграмм переходов в исполняемый код, а также практическая реализация этого метода в виде библиотеки STROBE на языке Ruby.]
- ⇒ ключевой знак*:
• автоматное программирование
• проектирование систем
- ⇒ оглавление*:
[
• Операционная семантика диаграмм переходов
• Реализация автоматных систем на языке Ruby
]
- ⇒ цитата*:
[Для проектирования и разработки реактивных систем часто используется SWITCH-технология, также известная как «автоматное программирование» или «программирование с явным выделением состояний». Одной из основных частей SWITCH-технологии является графический язык, позволяющий описывать поведение систем со сложным поведением в терминах состояний и переходов между ними и связей между этими системами. При использовании SWITCH-технологии в разработке программного обеспечения важной частью разработки является реализация поведения, описанного на диаграммах переходов, на целевом языке программирования.]

ЗАКЛЮЧЕНИЕ

В рамках учебной практики были изучены и повышены навыки формализации научных текстов. Были формализованы понятия, связанные с изучаемой дисциплиной и темой «Проблемы текущего состояния в области разработки и применения языков программирования», при помощи пакета макросов «scn-latex».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] В.С. Выхованец, О.Д. Выхованец. Параллельная парадигма программирования и её использование в языках высокого уровня / О.Д. Выхованец В.С. Выхованец. — Вестник Приднестровского университета, 1997. — Р. 74–79.
- [2] Гардейчик С.М., Корниевич А.И. Метапрограммирование как один из видов современных технологий программирования / Корниевич А.И. Гардейчик С.М. — БГПУ им. М. Танка.
- [3] Л.Н., Ясницкий. Интеллектуальные системы / Ясницкий Л.Н. — Лаборатория знаний, 2016. — Р. 221.
- [4] М.И., Шарикова. Иной взгляд на проблему «семантического разрыва» / Шарикова М.И. — Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2016.
- [5] М.П. Малыхина, Д.А. Герасимов. Мультиагентные системы искусственного интеллекта / Д.А. Герасимов М.П. Малыхина. — Научные труды КубГТУ, 2018. — Р. 476–484.
- [6] О.Г.Степанов,. Автоматное программирование с использованием динамических языков программирования / О.Г.Степанов. — Научно-технический вестник информационных технологий, механики и оптики.
- [7] Определение и проблемы языков программирования. — Московский государственный университет геодезии и картографии, 2016.
- [8] Созинова, Е. Н. Применение экспертных систем для анализа и оценки информационной безопасности / Е. Н. Созинова. — Молодой ученый, 2011. — Р. 64–66.
- [9] Что такое кроссплатформенность. Объясняем простыми словами. — <https://secretmag.ru/enciklopediya/cto-takoe-krossplatformennost-obyasnyаем-prostymi-slovami.htm>.