Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

по дисциплине «Интеллектуальные системы»

ЭКСПЕРТНАЯ СИСТЕМА ВЫБОРА СРЕДСТВА РАЗРАБОТКИ

Бондарев Матвей Владимирович

Копылов Данила Олегович

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) «Искусственный интеллект и разработка программных продуктов»

Руководитель работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Д. Брагин

*подпись*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Авторы работы

студент группы № 932204

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.В. Бондарев

*подпись*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

студент группы № 932101

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.О. Копылов

*подпись*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Томск – 2024

Оглавление

[ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1](#_heading=h.gjdgxs)

[ИДЕНТИФИКАЦИЯ ПРОБЛЕМНОЙ ОБЛАСТИ 3](#_heading=h.30j0zll)

[ПОСТАНОВКА ЗАДАЧИ 4](#_heading=h.1fob9te)

[РЕАЛИЗАЦИЯ ЭКСПЕРТНОЙ СИСТЕМЫ 5](#_heading=h.3znysh7)

[1.Построение модели экспертной системы. 5](#_heading=h.2et92p0)

[2.Реализация программного макета 6](#_heading=h.tyjcwt)

[3.Пример вывода в консоль результата работы программы 8](#_heading=h.3dy6vkm)

[ВЫВОД 9](#_heading=h.anyo6xujbt58)

# ИДЕНТИФИКАЦИЯ ПРОБЛЕМНОЙ ОБЛАСТИ

Данная экспертная система должна на основе вопросов о деятельности разработчика подобрать подходящую среду разработки.

Выделенные ресурсы:

* время;
* студенты;

Тип экспертной системы – прогнозирующая.

Назначение данной экспертной системы состоит в выборе подходящего пользователю редактора для облегчения процесса разработки.

# ПОСТАНОВКА ЗАДАЧИ

Цель: разработать экспертную систему для выбора средства разработки.

Задачи:

1. Построить модель данной экспертной области;
2. Реализовать код в заданной среде для основного алгоритма;
3. Продемонстрировать работу программы;
4. Оформить отчет в соответствии с требованиями методического описания.

Среда разработки: Microsoft Visual Studio Code.

# РЕАЛИЗАЦИЯ ЭКСПЕРТНОЙ СИСТЕМЫ

## 1.Построение модели экспертной системы.

Для начала был определен список вопросов, которое будут заданы пользователю для определения его предпочтений к редактору.

Список вопросов:

1. Для какой цели вам требуется редактор?
2. Какой язык вы используете в основном?
3. Готовы ли вы платить за использование редактора?
4. Какую операционную систему вы используете?
5. Нужна ли вам возможность интеграции с GIT?
6. Нужна ли вам возможность отладки приложений?

Список предполагаемых ответов:

* 1. Backend
  2. Data Science
  3. Game Development
  4. Просто требуется редактор
  5. Java
  6. Python
  7. PHP
  8. R
  9. Swift
  10. Да
  11. Нет
  12. Windows
  13. Linux
  14. MacOS
  15. Да
  16. Нет
  17. Не важно
  18. Да
  19. Нет
  20. Не важно

Экспертная система располагает следующими возможными вариантами ответа:

* IntelliJ IDEA
* PyCharm
* PhpStorm
* Jupyter Notebook
* RStudio
* Xcode
* Visual Studio Code
* Android Studio

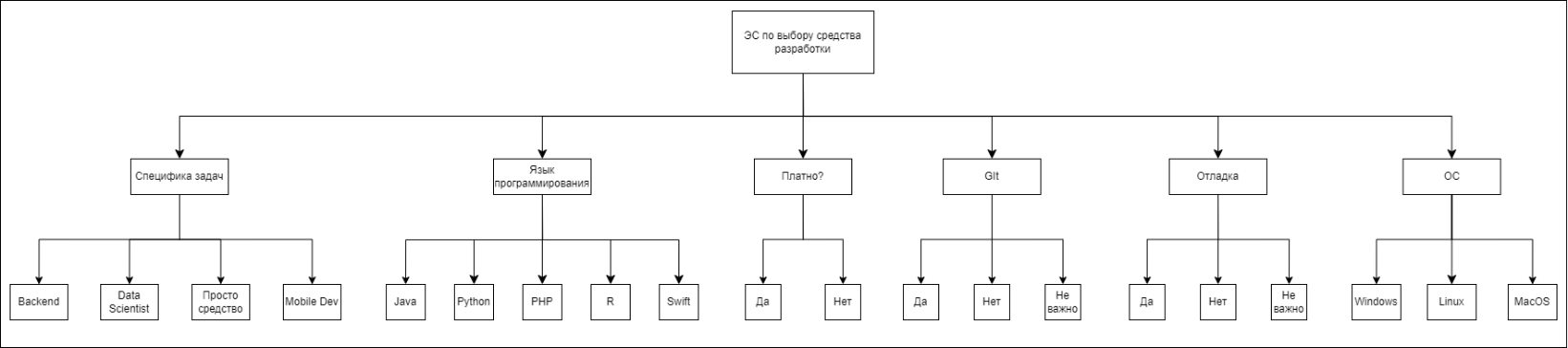


Рисунок 1 – Модель экспертной системы на графе

## 2.Реализация программного макета

В качестве механизма экспертной системы была выбрана схема Шортлиффа, она представлена в библиотеке языка Python: experta.

Приведем листинг программы:

class Solve(KnowledgeEngine):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.MDAndroidStudio = 0

self.MDIntelliJIDEA = 0

self.MDPyCharm = 0

self.MDPhpStorm = 0

self.MDJuNote = 0

self.MDRStudio = 0

self.MDXcode = 0

self.MDVSCode = 0

self.MNDAndroidStudio = 0

self.MNDIntelliJIDEA = 0

self.MNDPyCharm = 0

self.MNDPhpStorm = 0

self.MNDJuNote = 0

self.MNDRStudio = 0

self.MNDXcode = 0

self.MNDVSCode = 0

Выше приведен класс Solve, заданы атрибуты через конструктор по умолчанию, они отражают меру доверия и недоверия к определенному редактору

Далее в классе реализованы методы, отмеченные декораторами @Rule класса KnowledgeEngine, которые отвечают за настройку атрибутов доверия и недоверия в зависимости от ответа пользователя. Листинг этой части приводить не будем, так как он достаточно большой.

def getWeights(self):

return {

"IntelliJ IDEA": self.MDIntelliJIDEA - self.MNDIntelliJIDEA,

"PyCharm": self.MDPyCharm - self.MNDPyCharm,

"PhpStorm": self.MDPhpStorm - self.MNDPhpStorm,

"Jupyter Notebook": self.MDJuNote - self.MNDJuNote,

"RStudio": self.MDRStudio - self.MNDRStudio,

"Xcode": self.MDXcode - self.MNDXcode,

"Visual Studio Code": self.MDVSCode - self.MNDVSCode,

'Android Studio': self.MDAndroidStudio - self.MNDAndroidStudio,

}

Метод getWeight класса Solve возвращает словарь следующей структуры:

Название редактора: КУ

Где КУ – коэффициент уверенности, который получается при вычитании из меры доверия меры недоверия.

def checkRedactors(self):

dct = self.getWeights()

max\_key = max(dct, key=dct.get)

max\_value = dct[max\_key]

return max\_key, max\_value

Метод checkRedactors возвращает пару ключ МАКС\_КУ, что позволяет выбрать наиболее подходящий пользователю редактор.

Далее в программе представлены словари с вопросами ответами, однако они уже представлены выше, листинг этого места мы тоже опустим.

if \_\_name\_\_ == "\_\_main\_\_":

FindSolution = Solve()

for i in Questions:

print(Questions[i])

x = str(input(Answers[i]))

ans = str(i) + "-" + x

FindSolution.declare(Fact(answ=ans))

FindSolution.run()

check = FindSolution.checkRedactors()

print("Вам подходит редактор: " + check[0])

На фрагменте представлен код, который осуществляет запуск программы.

## 

## 3.Пример вывода в консоль результата работы программы

## 

Для какой цели вам требуется редактор?

1 - Backend

2 - Data Science  
3 - Game Development  
4 - Просто требуется редактор  
2  
Какой язык вы используете в основном?

1 - Java  
2 - Python  
3 - PHP  
4 - R  
5 - Swift  
2  
Готовы ли вы платить за использование редактора?

1 - Да  
2 - Нет  
2  
Какую операционную систему вы используете?

1 - Windows   
2 - Linux  
3 - MacOS  
1  
Нужна ли вам возможность интеграции с git?

1 - Да  
2 - Нет  
3 - Не важно  
3  
Требуется ли вам возможность отладки?

1 - Да  
2 - Нет  
3 - Не важно

3  
Вам подходит редактор: Jupyter Notebook

# ВЫВОД

В результате лабораторной работы была разработана экспертная система для помощи в выборе редактора, студенты научились составлять а также реализовывать алгоритм нечеткой логики, познакомились со схемой Шортлиффа и методом принятия решений на основе нечеткой логики.