Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №6**

По дисциплине «Проектирование баз знаний»

Тема: «Разработка логической базы знаний для автоматизации указанного вида деятельности»

**Выполнил:**

Студент 3 курса

Группы ИИ-21

Кирилович А. А.

**Проверил:**

Савонюк В.А.

Брест 2024

**Цель работы:** 1. Разработать логическую базу знаний для автоматического управляющего устройства.2. Получить навыки анализа предметной области и разработки логических баз знаний.

**Вариант 12**

|  |  |
| --- | --- |
| **12** | **Управление орошением в теплице**    Ваша программа должна описывать базу знаний для для управления поливом растений в теплице. Полив различен согласно типу растения:  Тип 1 для водяных растений в бассейне. Поплавковый выключатель всегда регулирует уровень воды в этом бассейне в заданных пределах.  Требуется, чтобы растения второго типа поливались 3 минуты под управлением временного выключателя в 06:00 утром и в 20:00 вечером.  Растения третьего типа поливаются каждые два дня вечером в течение 2 минут, когда выключатель сумерек активизирован. |

**Ход работы:**

**Решение:** Необходимо реализовать систему полива для каждого типа растений. Для первого реализуем постоянный полив с регуляризацией, которую сможет потом редактировать сам пользователь под конкретные нужды. Для второго типа необходимо реализовать полив в 6 и 20 часов в течении 3 минут. А для третьего необходимо реализовать полив каждые два дня во время сумерек в течении 2 минут.

**Входные переменные:**

I0 — сигнал поплавкового выключателя.

I1 — сигнал наступления времени 06:00.

I2 — сигнал наступления времени 20:00.

I3 — сигнал наступления сумерек.

**Выходные переменные:**

Q1 — Включение полива.

Q2 — Выключение полива.

**Переменные времени:**

T1 — достижение времени полива для второго типа растений.

T2 — достижение времени полива для третьего типа растений.

Т3 — достижение времени с последней активации для третьего типа растений.

Далее — разобьем нашу задачу на условные части и составим логические высказывания для каждой из частей.

1. Для первого типа растений:
   1. Включаем полив: Q1 = I0
   2. Выключаем полив: Q2 = not(I0)
2. Для второго типа растений:
   1. Включаем полив: Q1 = I1
   2. Выключаем полив: Q2 = not(Т1)
   3. Включаем полив: Q1 = I2
   4. Выключаем полив: Q2 = not(Т1)
3. Для второго типа растений:
   1. Включаем полив: Q1 = I3 \* Т3
   2. Выключаем полив: Q2 = not(Т2)

**Код программы:**

import threading  
import time  
import datetime  
from astral import LocationInfo  
from astral.sun import sun  
from pytz import timezone  
  
city\_name = "Москва"  
latitude = 55.7558  
longitude = 37.6176  
timezone\_str = "Europe/Moscow"  
location = LocationInfo(city\_name, "Russia", timezone\_str, latitude, longitude)  
  
  
def get\_time\_of\_dawn\_and\_dusk():  
 current\_time = datetime.datetime.now(timezone(timezone\_str))  
  
 s = sun(location.observer, date=current\_time.date())  
  
 return s["dusk"].astimezone(timezone(timezone\_str))  
  
  
class WateringSystem:  
 def \_\_init\_\_(self):  
 self.water\_level = 0  
 self.last\_watering\_time = datetime.datetime.now()  
 self.lock = threading.Lock()  
  
 def manage\_water\_level(self):  
 with self.lock:  
 print("Полив начался.")  
  
 def water\_for\_duration(self, minutes):  
 with self.lock:  
 print(f"Полив на {minutes} минут")  
 time.sleep(minutes \* 60)  
 self.update\_last\_watering\_time()  
  
 def update\_last\_watering\_time(self):  
 self.last\_watering\_time = datetime.datetime.now()  
  
  
def water\_plants(plant\_type, watering\_system):  
 while True:  
 current\_time = datetime.datetime.now(timezone(timezone\_str))  
  
 if plant\_type == 1:  
 watering\_system.manage\_water\_level()  
 elif plant\_type == 2:  
 if current\_time.hour == 6 or current\_time.hour == 20:  
 watering\_system.water\_for\_duration(3)  
 elif plant\_type == 3:  
 dusk\_time = get\_time\_of\_dawn\_and\_dusk()  
 if current\_time - dusk\_time > datetime.timedelta():  
 if (current\_time - watering\_system.last\_watering\_time).days >= 2:  
 watering\_system.water\_for\_duration(2)  
 time.sleep(60)  
  
  
watering\_system = WateringSystem()  
  
# thread\_1 = threading.Thread(target=water\_plants, args=(1, watering\_system))  
# thread\_2 = threading.Thread(target=water\_plants, args=(2, watering\_system))  
thread\_3 = threading.Thread(target=water\_plants, args=(3, watering\_system))  
  
# thread\_1.start()  
# thread\_2.start()  
thread\_3.start()  
  
# thread\_1.join()  
# thread\_2.join()  
thread\_3.join()

**Вывод:** 1.Разработал логическую базу знаний для автоматического управляющего устройства. 2. Получил навыки анализа предметной области и разработки логических баз знаний.