**Системы искусственного интеллекта**

1 блок

1. Define different fields that the Artificial intelligence covers. Describe main tasks of the Artificial intelligence.
2. Define different fields that the Artificial intelligence covers. Write about natural language processing and knowledge representation.
3. Define different fields that the Artificial intelligence covers. Write about automated reasoning, machine learning and robotics.
4. Intelligent agents. Describe the approach of the rational agent. Give examples of rational and irrational situations.
5. Intelligent agents. Describe performance, environment, actuators and sensors of agents.
6. Intelligent agents. Write about the structure of agents and agent programs.
7. Intelligent agents. Describe how an agent interacts with the environments. You need to focus on sensors and actuators.
8. Intelligent agents. Describe a rational agent. Give an example of the specific work of the rational agent.
9. Intelligent agents. Describe an irrational agent. Give an example of the specific work of the irrational agent.
10. Search algorithms. Describe the ways that can be used for a search in arrays, lists and linked lists.
11. Search algorithms. Describe the ways that can be used for a search in graphs and binary trees.
12. Search algorithms in binary trees. Describe the Breadth-first search. Write down an algorithm and draw its picture.
13. Search algorithms in binary trees. Describe the Depth-first search. Write down an algorithm and draw its picture.
14. The adversarial search. Describe the difference between standard search algorithms and an adversarial search algorithm.
15. The adversarial search. Describe its features. Write down where an adversarial search could be efficiently used (for example, games).
16. Describe the Minimax algorithm on an example of the tic-tac-toe game.
17. Describe the Minimax algorithm on an example of a multiplayer game with the tree that has nodes of different weights.
18. Describe the Minimax algorithm on an example of a multiplayer game with the tree that has nodes of different weights. Add alpha-beta pruning to the problem.
19. The Wumpus world game. Draw the scheme and write down the rules of the game.
20. The Wumpus world game. Write about the Wumpus-world game and its environments, actuators (actions we do in the game) and sensors (Stench, Breeze, Bump, Scream).

2 блок

1. Describe logical agents and knowledge-based agents.
2. Describe logical entailmentbetween sentences α |= β. You can use the Wumpus-world cells as examples of this problem.
3. Propositional logic. Describe its syntax and semantics.
4. Propositional logic. Write down examples of atomic and complex sentences.
5. Propositional logic. Describe connectors of complex sentences. Give examples of complex sentences with each of these connectors.
6. The first-order logic. Describe different models for the first-order logic.
7. The first-order logic. Write about predicates, atomic and complex sentences; describe the ways that are used to create them.
8. The first-order logic. Write about connectors of complex sentences. Give examples of complex sentences using different connectors.
9. The first-order logic. Write about quantifiers you know. Give examples of sentences using these quantifiers.
10. The first-order logic. Describe states of the Wumpus world using the first-order logic predicates.
11. Describe the inference in the first-order logic. Give examples of sentences with the inference.
12. Give examples of the first-order definite clauses.
13. Describe a simple forward-chaining algorithm. Match rules against known facts. Give examples.
14. Describe a backward-chaining algorithm. Give examples.
15. Describe the constraint logical programming. Give examples.
16. Describe classical planning problems like air cargo transport, the spare tire problem and the blocks world.
17. Describe the knowledge representation of categories and objects including physical compositions, measurements and things and stuff.
18. Describe the meaning representation using events. Give examples.
19. Describe the meaning representation using events. Write about time intervals. Give examples of events with time intervals.
20. Describe mental events and mental objects in the knowledge representations. Give examples.

3 блок

1. Create a program in Visual Prolog. The database contains the following facts:

likes ("Anna," apples).
likes ("Sergey", bananas).
likes ("Andrey", apples).
likes ("Sveta", chocolate).
likes ("Vova", chocolate).
likes ("Anna", chocolate).
loves ("Sveta", oranges).
likes ("Vova", bananas).

Make a program that defines:
a) everyone who loves bananas;
b) who likes both chocolate and apples;
c) what Vova likes;
d) that they love both Sveta and Vova.

1. Create a program in Visual Prolog. The database contains the following facts:
plays ("Sasha", football).
plays ("Katya", tennis).
plays ("Sasha", tennis).
plays ("Andrew", football).
plays ("Oleg", football).
plays ("Olga", tennis).
plays ("Katya", volleyball).
plays ("Oleg", volleyball).
Make a program that defines:
a) what kind of sport is Andrey interested in;
b) everyone who plays volleyball;
c) what kind of sport are Olga and Sasha interested in;
d) who is fond of football and volleyball.
2. Create a program in Visual Prolog. The database contains the following facts:
likes ("Anna," apples).
likes ("Sergey", bananas).
likes ("Andrey", apples).
likes ("Sveta", chocolate).
likes ("Vova", chocolate).
likes ("Anna", chocolate).
loves ("Sveta", oranges).
likes ("Vova", bananas).
fruit (apples).
fruit (bananas).
fruit (oranges).
sweets (chocolate).
a) using the available facts, to compose a new rule for any fruits (X) and identify all those who like fruit;

b) using the available facts, create a new rule for any candy (X) and identify everyone who loves sweets;
c) using the available facts, make a rule of any taste (X) and identify everyone who likes fruits and sweets.

1. Create a program in Visual Prolog. The database contains the following facts:
fondOf ("Kolya", “guitar”).
fondOf ("Olya", “violin”).
fondOf ("Dima", “swimming”).
fondOf ("Tanya", “tennis”).
sport (“swimming”).
sport (“tennis”).
music\_inst (“violin”).
music\_inst (“guitar”).
a.) Make up the athlete's rule and determine who is keen on sports;

b.) Make up a musician rule and determine who is fond of music;

1. Create a program in Visual Prolog. The database contains the following facts

builds (“Gim”, “lumber\_mill”).
builds (“Sylvanas”, “smithy”).
builds (“Ankeron”, “tower”).
builds (“Druko”, “workshop”).
builds (“Garrosh”, “fortress”).

builds (“Cronosh”, “tower”).

builds (“Alleria”, “smithy”).
team1 (“Gim”).

team1 (“Druko”).

team1 (“Cronosh”).

team2 (“Garrosh”).

team2 (“Alleria”).

team2 (“Sylvanas”).

team2 (“Ankeron”).
Make a program that defines:
a) who builds the workshop;
b) everyone who builds the tower;
c) anyone who is in a team 1;

d)create a new rule build\_team(X)and identify anyone who builds the smithy and in a team 2

1. Create a program in Visual Prolog. The database contains the following facts

research (“Gally”, “space”).
research (“Gnilias”, “medicine”).
research (“Odin”, “armory”).
research (“Dave”, “nuclear\_weapon”).
research (“Chen”, “cookery”).

research (“Rango”, “medicine”).

research (“Varimatras”, “space”).
squad1 (“Gnilias”).

squad1 (“Odin”).

squad1 (“Chen”).

squad2 (“Dave”).

squad2 (“Rango”).

squad2 (“Gally”).

squad2 (“Varimatras”).
Make a program that defines:
a) who conducts research about armory;
b) everyone who conducts research in space;
c) anyone who is in a squad 1;

d)create a new rule research\_team(X) and identify anyone who conducts research in space and in a squad 2

1. Create a program in Prolog for calculating the value of the expression Y = (X2 + 1) / (X-2) for the variable X. Create a program for calculating the value of the expression S = 2 (X2 + Y2) / (X + Y) for the variables X and Y. Do not forget to check for allowable values
2. Create a program in Prolog for calculating the value of the expression z = sinx + 3lnx for the variable X. Create a program for calculating the value of the expression y = ln (sinx + ex) for the variable X.
3. Write down a program in Prolog that finds the result of the following equation Z=(X+2\*Y)/ln(X) for input numbers X and Y. Do not forget to check for allowable values. Write down a program in Prolog that finds the result of the following equation Z=(X-Y) (Y+X) / (2\*X - Y) for input numbers X and Y. Do not forget to check for allowable values.
4. Make a program in Prolog to check whether the entered number X is within the specified interval [a, b].
5. Make a program in Prolog to select the smallest of the three entered numbers. Make a program in Prolog to select the largest of the three entered numbers.
6. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the sequence of numbers using the Breadth-first and Depth-first search in the binary tree



1. Write down the function that will realize the Breadth-first search for the following class of the binary tree

class Node {

 int data;

 Node \*left;

 Node \*right;

 public:

 Node() { data = -1; left = NULL; right = NULL; };

 void setKey(int key) { data = key; };

 void setLeft(Node\* Left) { left = Left; };

 void setRight(Node\* Right) { right = Right; };

 int getKey() { return data; }

 Node \*getLeft() { return left; }

 Node \*getRight() { return right; }

}

1. Write down the function that will realize the Depth-first search for the following class of the binary tree

class Node {

 int data;

 Node \*left;

 Node \*right;

 public:

 Node() { data = -1; left = NULL; right = NULL; };

 void setKey(int key) { data = key; };

 void setLeft(Node\* Left) { left = Left; };

 void setRight(Node\* Right) { right = Right; };

 int getKey() { return data; }

 Node \*getLeft() { return left; }

 Node \*getRight() { return right; }

}