

## **BPI1235 (BPI1239) Mesterséges intelligencia**

### **Tantárgyi tematika és félévi követelményrendszer**

**Heti kontakt óraszám (elm.+gyak.): 2+2**

**Előfeltétel (tantárgyi kód) BPI1204 (programozás 2)**

**Oktató: Vályi Sándor**

- *elektronikus levelezőcím: [valyi.sandor@nye.hu](mailto:valyi.sandor@nye.hu), 2 munkanapi reakcióidő, nem azonnali ügyfélszolgálat!*
- *A kurzushoz tartozik: [classroom.google.com](https://classroom.google.com)- Felvételi kulcs: os43fma itt fogunk gyakorló feladatokat megoldani interaktívan, de jelenléti módon, a tanteremben.*

**Félévi követelmény:** vizsga (évközi alapján lehetséges jegymegajánlás is)

**Évközi követelmény, vizsgára bocsájtás feltétele:**

beadandó program készítése és védeése [40 p]

2 darab papíron írt dolgozat [45 p + 15 p]

A minimum követelmény 30 p elérése, ezzel lehet menni szóbelileg vizsgázni. 40 ponttól elégséges jegymegajánlás, 60-tól közepes, 75-től jó, 90-től jeles.

Az évközi követelményeket egyszer lehet csak ismételni várhatóan június elején, ezek alapján már nem kötelező a jegymegajánlás az oktató számára.

**A vizsga: szóbeli.** Itt a jegymegajánlás és az évközi pontok már nem számítanak, minden jegyet lehet szerezni 1..5 intervallumban.

**Féléves tematika:**

**Az órák programja (tematika):**

<b>Hét</b>	<b>Előadáson</b>	<b>Gyakorlaton</b>	<b>Dátum</b>
<b>1</b>	Bevezetés, intelligens ágensek, generatív mestint használata programozásban	Ágensek tulajdonságai; Egyszerű ágensek programozása, programozás és tesztelés chatGPT aktuális ingyenes verziójával	<b>febr. 21</b>
<b>2</b>	Problémák állapottér-reprezentációja	Példák állapottér-reprezentációra, Java keretrendszer állapottér-reprezentált problémák számára	<b>febr. 28</b>

3	Problémamegoldás, mint útkeresés	Mélységi, szélességi és egyéb nem informált keresők.	<b>márc. 6</b>
4	Informált keresési eljárások	Informált keresők	<b>márc. 13</b>
5	2-személyes játékok, Java keretrendszer a 2-személyes játékokhoz. Példa a beadandó 0-1-2. produktumára	Minimax algoritmus, alfa-béta vágás. Beadandó-témaválasztás.Beadandó 0. feladat.	<b>márc. 20</b>
6	Adatbázist nem építő keresési eljárások. Genetikus algoritmusok, backtrack eljárás. Kényszerfeltételek az állapotokon.	Beadandó 1. produktum, példa dolgozat	<b>márc.27</b>
7	ZH-előkészítő, <b>elméleti dolgozat</b>	<b><u>gyak ZH I.</u></b>	<b>ápr. 3</b>
8	Beadandó példaprogram bemutatása	Beadandó 2. produktum elkészítése	<b>ápr. 10</b>
9	Nulladrendű és elsőrendű logika és ismeretreprezentáció, következtetés, automatikus következtetés: rezolúció ítéletlogikára	Ismeretreprezentáció Tarski világában. Logikai feladatok megoldása -- tutorial.	<b>ápr.17</b>
10	Logikai programozás, Prolog, kényszerfeltételes logikai programozás. Szakértői és döntéstámogatási rendszerek. Ismeretreprezentációs módszerek. Szakértői rendszerek és logikai programozás.	Prolog gyakorlatok. Beadandó konzultáció.	<b>ápr.24</b>
11	Bizonytalan ismeretek ábrázolása, pl. Bayes-hálók. Tanuló algoritmusok, Döntési fák.	Gyakorlatok a Bayes-hálókról, ID3 algoritmusról, döntési fákrol	<b>máj. 1 pótlása</b>
12	Neuronhálók, deep learning a szövegfeldolgozásban	Beadandó védés.	<b>máj. 8</b>

13 ZH II. dolgozat az előadáson

ZH II folytatása

máj.15

14 Tartalék időpont

Tartalék időpont

máj.22

### ***Értékelési rendszer (évközi és vizsga)***

-- **évközi 1. követelmény:** 40 pont **beadandó**, ami a későbbiekben részletesen le van írva.

Ennek benyújtása messze nem elég: véde a gyakorlatokon, ez, ami számít.

A beadandóválasztás az ötödik hétig a moodle-n. A beadandó köztes állapotainak bemutatása szintén, a gyakorlatokon. Benyújtási határidő a táblázatban. Minimum pontszám 24, anélkül nem lehet vizsgázni.

<i>Fázis</i>	<i>Leírás</i>	<i>Hét</i>	<i>Pontszám</i>
0. <i>fázis</i>	a feladat szöveges leírásának feltöltése	5	0
1. <i>fázis</i>	a választott témák problémáinak állapottér-reprezentációja "papíron", állapotok, operátorok leírása egy dokumentumban. A dokumentáció tartalmazza az állapottér (játékfa) egy 10-12 állapotú részének ábráját, ahol a lehetséges operátorok is elmagyarázhatók, valamint azt, hogy milyen kereső lesz leprogramozva.	6	15
2. <i>fázis</i>	állapottér-repr. Javában programozva is, keresőalgoritmus beállítása, működik a keresés, tesztek bizonyítják, hogy jól működik. A program az állapottérben való keresés egyes lépéseit is írja ki, ne csak a megoldást.	12	25

### **A beadandóról:**

A beadandó program problémamegoldója az állapottér gráfrepresentációján kell alapuljon és valamely, a kurzuson tanult keresőalgoritmust felhasználva (mélységi, szélességi kereső etc. avagy minimax algoritmus, alfa-béta vágás) kell működnie.

A problémamegoldás/játék inicializálható legyen egy kezdőállással, amit input fájlból olvasunk föl. 2 személyes játék esetén az egyik ellenfél lépéseit humán vezesse, a másik fél lépéseit a minimax lépésajánló, legalább 8 féllépés mélységig.

Előny, ha a program grafikus felhasználói felülettel rendelkezik, pályaszerkesztési funkcióval (a kezdőállás grafikus megszerkesztése). Előny, ha annyi távoli adatbázis-használatot tartalmaz, hogy elmenti az adott felhasználó nevét, a megszerkesztett generált pályát egy megadott névvel, a játszmát és az eredményt, vagy relációs adatbázisban, vagy XML-adatként. Mondani sem kellene, vissza is kell tudnia tölteni a mentésből a név alapján kikeresve a több elmentett állás közül.

A bemutatók feladatokban való mély ismereteikről személyes védés során számolnak be. Az elkészített programban meg kell jelölni azon kódrészeket, amelyeket nem a védő készített. Olyan kérdések is várhatók, hogy mit kellene másképp csinálni, ha változtatni akarnánk a program működésén.

Kivétel a fent leírt beadandó-követelmények alól:

- aki tanuló ágenszt készít open source könyvtárral adott inputból adott intervallum-értékeket előállító intervallum-értékű számítások előállítására, pl. 'Deep Learning' neuronhálókat alkalmazva, ő jeles megajánlott jegyet kap, s sejtethetően mehet vele országos TDK-ra is.
- aki működő ágenszt készít reális (amely problémákat szokták mellékelni a PC-s sokobanhoz) Sokoban feladatok megoldására, jeles megajánlott jegyet kap.
- aki olyan webhelyet készít, pl. Java web service, Spring MVC (ez a preferált!) technológiára, amelyre a regisztrált userek felvihetnek Sokoban álláskiértékelő heurisztikát visszaadó int heur(SokobanState) metódus szövegét, Java-ban megírva, és a webhely eme heurisztikát injektálva az A\* keresőbe keres megoldást a szintén a userek által felvitt pályákra, és időszakonként rangsort hirdet a userek heurisztikái között, a teljesítményük alapján, ő jeles megajánlást kap.
- más, saját ötletes, neuronhálós vagy más gépi tanulást tartalmazó feladatmegoldás, ami példaként bemutatható a csoporttársaknak.

### **Az érdemjegy kialakításának módja:**

*A teljesítménnyel pozitív korrelációban, a pontszámok alapján..*

### **Oktatási segédanyag:**

Az előadáson bemutatott prezentációk megtalálhatók a <http://moodle.nye.hu> címen.

**Kötelező<sup>1</sup> és ajánlott irodalom:**

- Futó Iván (szerk.): Mesterséges intelligencia, Aula Kiadó, 1999.
- Fekete István, Gregorics Tibor, Nagy Sára: Bevezetés a mesterséges intelligenciába, LSI Oktatóközpont, 1990, ELTE Eötvös Kiadó, 2006.

**Stuart J. Russell, Peter Norvig<sup>1</sup>: Mesterséges intelligencia modern megközelítésben, Panem, Budapest, 2005. (1-9. fejezet a kötelező, a 2003-as második kiadás fordítása) ,**  
Online elérhető, URL:

<https://dtk.tankonyvtar.hu/handle/123456789/8703>