# **Утвърдил: …………………..**



# **/** ..................................... /

Утвърден от Факултетен съвет   
с протокол № ............. / ...............

# **СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”**

# Факултет по Математика и Информатика

Специалност: *Информатика, Информационни системи, Компютърни науки, Математика и информатика, Софтуерно инженерство*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

Курс: 2+

Учебна година: 2018/2019

Семестър: IV

# УЧЕБНА ПРОГРАМА

##### **Дисциплина: Практическа роботика и умни "неща" /**

**Practical Robotics and Smart Things**

Избираема дисциплина

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

### Преподавател: ас. Траян Илиев

Асистент: ас. Траян Илиев

|  |  |  |
| --- | --- | --- |
| **Учебна заетост** | **Форма** | **Хорариум** |
| Аудиторна заетост | Лекции | 30 |
| Семинарни упражнения |  |
| Практически упражнения (хоспетиране) | 30 |
| **Обща аудиторна заетост** | | **60** |
| Извънаудиторна заетост | Курсов учебен проект | 50 |
| Доклад/Презентация | 10 |
| Подготовка на домашни работи | 10 |
| Подготовка за финален тестови изпит | 10 |
| Самостоятелна работа с ресурси | 10 |
| **Обща извънаудиторна заетост** | | **90** |
| **ОБЩА ЗАЕТОСТ** | | **150** |
| **Кредити аудиторна заетост** | | **2** |
| **Кредити извънаудиторна заетост** | | **3** |
| **ОБЩО ЕКСТ** | | **5** |

|  |  |  |
| --- | --- | --- |
| **№** | **Формиране на оценката по дисциплината** | **% от оценката** |
|  | Финален курсов проект | 40% |
|  | Финален тестови изпит | 40% |
|  | Домашни работи през семестъра (текущ контрол) | 10% |
|  | Работа в час (задачи по време на упражнения) | 10% |
| **Анотация на учебната дисциплина:** | | |
| Интелигентните устройства са навсякъде около нас – автомобилите и домовете стават „умни“, дрехите включват „интелигентни материи“ и незабележимо вградена електроника (*wearable electronics*), почти всички електроуреди и предмети на бита придобиват способности за комуникация помежду си и с отдалечени услуги „в облака“ (*cloud computing*). Често взаимодействието става с гласови команди (*Amazon Alexa*). В крайна сметка цялата тази информация и възможност за контрол стават достъпни за нас през удобен вграден, мобилен или гласов интерфейс от всяко място, по всяко време (*consumer application dashboards*).  С помощта на множество практически проекти, курсът запознава с бързо-развиващата се област на „Интернет на нещата“ (*Internet of Things – IoT*), обслужващата и социална роботика (*service and social robotics*). Акцентът е върху придобиване на реален опит при реализация на вградени и мрежово свързани устройства и малки роботи (*Arduino, Raspberry Pi 2/3/Zero, ESP 8266, Lego, сензори, актуатори* и др.) и най-вече върху начина, по който тези устройства комуникират помежду си и със заобикалящия ги свят (*социална способност*).  За да реализираме на практика тези проекти се нуждаем от рамка за моделиране на взаимодействието между устройствата, разглеждани като *интелигентни агенти (Intelligent Agents - IA)* в рамките на една *многоагентна система (Multi-Agent System - MAS)*. Разглеждат се основните характеристики на интелигентните агенти - автономност, способност за реагиране, проактивност, способност за самообучение (адаптивност), социална способност (езици за комуникация между агенти - ACLs), рационалност, мобилност, както и различни парадигми и архитектури за реализация – йерархична (базирана на планиране), реактивна (*Subsumption Architecture*), и хибридна. Включено е запознаване с *Belief-Desire-Intention (BDI)* модел на практически разсъждения като основа за моделиране на света, комуникация и автономно вземане на решения от агентите. Знанията се представят и реферират с помощта на онтологии и *Semantic Web* *W3C* стандарти (*RDF/RDFS, OWL*).  Курсът изгражда цялостна картина на необходимите технологии започвайки от хардуерния слой (практически проекти с *Arduino, Raspberry Pi* и *ESP 8266 +* най-различни сензори и актуатори: ултразвукови и инфрачервени за дистанция, оптично масиви за следене на линия, камери, енкодери, двигатели и драйвери, серво- механизми за хващане на предмети и движение на камерата, сензорни *TFT* екрани и други), през обработката на събитийни потоци в реално време с помощта на микро-контролери и серийна *USB (UART), I2C, SPI*, комуникация, софтуерни библиотеки от ниско ниво (*Java:* *LeJOS, Pi4J*), до горните слоеве на приложението за *реактивна обработка на събития* реализирани със *Spring Reactor*, *Akka*, *Eclipse IoT* (*Kura, Paho, Californium*) платформи.  Следващите нива в *IoT/ Service Robotics* архитектурата включват механизми и протоколи за комуникация - *Constrained Application Protocol (CoAP), MQ Telemetry Transport (MQTT), HTTP, уеб услуги* (*REpresentational State Transfer - REST APIs)*, "*облачни технологии" (Docker, Kubernetes, Apache Brooklyn, Ansible)*, бази от данни в реално време (*Prometheus, Graphite, InfluxDB, RethinkDB*).  Включени са също така и технологии за изграждане на вградени (*Java Swing, JavaFX*) и мобилни (*Angular + TypeScript + Material Design / Ionic*) уеб интерфейси и *интерактивни "табла за управление" (dashboards),* както и технологии за визуализация и анализ на събития в реално време (*Grafana).*  Целта е освен практическите проекти реализирани по време на упражнения, участниците да сформират екипи за реализиране на собствени роботи/ IoT много-агентни системи и/или компоненти за тях, които да бъдат демонстрирани в края на курса. Дългосрочната цел е сформиране на екип за участие в национални и международни състезания от типа на ***RoboCup*** (<http://www.robocup.org/>), където роботи изпълняват задачи с различна степен на сложност – играят футбол, спасяват хора при бедствия и аварии, кооперират се с хора за изпълнение на задачи у дома, в промишлеността и логистиката.  Информация за някои от роботите, които ще сглобяваме и програмираме по време на курса можете да намерите на адрес: <http://robolearn.org/>.  Кодът е достъпен в GitHub: <https://github.com/iproduct/course-social-robotics> | | |
| **Предварителни изисквания:** | | |
| Очаква се студентите да могат да боравят свободно с технически английски език. Необходимо е добро познаване на езика Java, желателно и на езика Python. Препоръчителна, но не задължителна, е също базова техническа грамотност за работа с цифрова схемотехника. | | |
| **Очаквани резултати:** | | |
| По време на курса студентите ще придобият знания и умения за:   * многоагентните системи с разпределен изкуствен интелект, кооперативната, социалната и домашната роботика и автоматизация; * обектно-ориентирано, актьор-ориентирано и агентно-ориентирно софтуерно архитектурно моделиране,*реактивно програмиране (reactive programming)* и *actor model* –***Spring******Reactor, RxJava, RxJS, Akka***; * реализация на вградени интерфейси за *IoT* устройства и роботи с ***Java Swing* /*Java FX***  на *ARM Linux* платформа; * реализация на мобилни уеб/ *native* интерфейси за *IoT* устройства и роботи с ***Angular 5, TypeScript, Material Design / Ionic****;* * практически опит с платформи ***Raspberry Pi 3, Lego® Mindstorms EV3*** и ***Arduino*** за изграждане на малки *роботи* и *IoT* устройства; * свързване на сензори (*ултразвукови и инфрачервени за дистанция, оптично масиви за следене на линия, камери, енкодери*), актуатори (*двигатели и драйвери, серво-механизми за хващане на предмети и движение на камерата*), вградени интерфейси (сензорни *TFT* екрани - *touchscreens*), и комуникационни модули *(WiFi, BlueTooth Low Energy - BLE, ESP 8266 System-On-Chip****)***; * ***IoT*** архитектури и ***протоколи за комуникация*** (***MQTT, CoAP, AMQP***) и конфигуриране на устройства (***OMA-DM, LWM2M***); * практическа реализация на *IoT Greenhouse* проект с ***Eclipse IoT*** платформа (*Kura, Paho, Californium* – *MQTT* и *CoAP* комуникационни протоколи); * Визуализация на данни в реално време с ***Grafana***; * *Java®* програмиране на ***Lego®*** роботи с използване на библиотеката ***LeJOS;*** * *Java®* програмиране на ***Raspberry Pi 3*** роботи и мрежово-свързани, вградени устройства с използване на библиотеката ***Pi4J;*** * управление сензори и актуатори през ***GPIO*** интерфейс – протоколи: ***Serial UART, I2C, SPI***; * програмиране на ***Arduino*** хардуер, сензори, актуатори и комуникационни протоколи с използване на ***Arduino IDE / Eclipse (C++);*** * обработка на изображения и компютърно зрение с библиотеката ***OpenCV***; * планиране на действия и манипулиране на обекти със ***STRIPS*** *(Stanford Research Institute Problem Solver);* * ***Belief–Desire–Intention (BDI)*** модел на човешките практически разсъждения и езици за комуникация между интелигентни агенти; * представяне на знания с помощта на онтологии и W3C стандарти за изграждане на ***Семантична Мрежа (Semantic Web):*** *Resource Description Framework (RDF)* и *Web Ontology Language (OWL)*.   Студентите ще изградят също умения за целенасочено търсене, анализ и употреба на информация, както и умения за работа в екип по избран от тях практически *IoT / Robotics* проект. | | |

#### *Учебно съдържание*

|  |  |  |
| --- | --- | --- |
| **№** | **Тема:** | **Хорариум** |
|  | Представяне на курса, въведение в областта на „Интернет на нещата“ *(Internet of Things – IoT)*, обслужващата и социална роботика (*service and social robotics*), умни домове (*smart homes*), автономни транспортни средства (*autonomous vehicles*), интелигентни материи (*smart fabrics* *& wearable electronics*), агрегация и обработка на сложни събития (*Complex Event Processing - CEP*) в облака (*cloud computing*). Разпределена обработка на потоци от събития в реално време. Ламбда архитектура: *Apache Spark, Storm, Kafka, Apex, Flink, Beam*. Многослойна референтна архитектура на *IoT*. *Platform as a Service (PaaS)*. Наблюдение и управление на *IoT* мрежи, устройства и роботи чрез вградени и мобилни уеб/ *native* интерфейси *–* *application dashboards* с *Grafana*. | 4 |
|  | Приложение на многоагентните системи с разпределен изкуствен интелект в кооперативната, социалната и домашната роботика и автоматизация. Обектно-ориентирано, актьор-ориентирано и агентно-ориентирано софтуерно инженерство. Реактивно програмиране (*reactive programming*) и *actor model* – *Spring Reactor, RxJava, RxJS, Akka*. Отличителни характеристики на интелигентните агенти. Видове интелигентни агенти. Интелигентни агенти с хардуерна архитектура – роботика. Колаборативна и социална роботика. Хардуерни платформи за изграждане на малки роботи и IoT устройства – запознаване с *Raspberry Pi 2, Lego® Mindstorms EV3* и *Arduino*. | 4 |
|  | Рационални агенти – *AIMA (Artificial Intelligence: Modern Approach)* агенти. Структура на рационални агенти. Агенти базирани на таблица. Агенти с прости рефлекси. Агенти с вътрешно състояние. Агенти базирани на цели. Агенти базирани на полезност. Агентни среди – видове и свойства. Програмна реализация на агенти и агентни среди. Примери. | 4 |
|  | Изграждане на роботи с *Lego® Mindstorms.* Програмиране на *Lego®* роботи. *Java®* програмиране на *Lego®* роботи с използване на *LeJOS – Wifi* конфигуриране, разработка с *LeJOS*, билд файлове, използване на сензори, мотори, инструменти,  *ssh* достъп до *EV3* блока, примерни програми и практически упражнения с *LeJaRo* робот и *LeJOS* – избягване на препятствия, хващане на предмети, следване на линия. | 4 |
|  | Програмиране на *Raspberry Pi 3* роботи и IoT устройства с използване на Java библиотеката *Pi4J*. Управление на сензори и актуатори през *GPIO* интерфейс – основни протоколи за комуникация: *Serial, I2C, SPI*. Елементи на схемотехниката – аналогово-цифрови преобразуватели, *level shifters* и др. Сглобяване на малък роботс *Raspberry Pi 3.* Управление на драйвера на двигателите през *GPIO* с *Pulse Width Modulation (PWM).* Добавяне на сензори за дистанция и програмиране на прости рефлекси у *Raspberry Pi* робота. | 4 |
|  | Хардуерна и софтуерна платформа с отворен код *Arduino –* хардуер, сензори и актуатори, комуникационни протоколи, езици за програмиране, среди за разработка *Arduino IDE / Eclipse (C++)*. Сглобяване и програмиране на примерни *Arduino* проектисъс сензори и актуатори. | 4 |
|  | *Интернет на нещата (IoT)* - практически проекти с *ESP 8266 WiFi SOC (System-On-Chip)* модули с интегриран *TCP/IP* стек от протоколи. *IoT* архитектури и *протоколи за комуникация* (*MQTT, CoAP, AMQP*) и конфигуриране на устройства (*OMA-DM, LWM2M*). *Eclipse IoT* *платформа*, проекти: *Edje, Paho, Wakaama, Kura, OpenHAB/SmartHome, Californium, Mosquitto, Leshan, Hono, hawkBit, BIRT*. Реализация на примерен проект *Greenhouse Tutorial* с *Eclipse Kura, Paho* и *Californium*. | 4 |
|  | Взаимодействие и кооперация между хора и интелигентни агенти с хардуерна архитектура (роботи). Проектиране и реализация на графични интерфейси за вградени устройства и роботи *с Java Swing /* *Java FX*  на *ARM Linux* платформа. | 4 |
|  | Разработка на уеб интерфейси за управление на *IoT* устройства с *JavaScript* библиотеките *Angular 2, TypeScript, Material Design. Angular CLI. Ionic 2* платформа за разработка на *native* мобилни приложения с *Angular* и *TypeScript*. | 4 |
|  | *Akka toolkit* за разработка на масивно конкурентни, разпределени и устойчиви към грешки събитийно-ориентирани *актьор (аctor) системи* – референции, състояние, поведение, наследници, супервайзинг стратегия, прекъсване. Наблюдение и контрол – стратегии *One-For-One* и *All-For-One*. Делегиране на задачи на наследници. Референции, пътища и адреси на актьори. Селекция по път. Обработка на изключения и грешки. Жизнен цикъл на актьор – *DeathWatch*. Изпращане и получаване на съобщения – *tell, ask, reply*. Спиране на актьори. Динамично добавяне/замяна на поведения на актьори – *become* and *unbecome* (шаблон *State*). Актьори със *stash*. Диспечери на съобщения. *Mailboxes* – типове конфигурация и приоритети. Маршрутизация на съобщения и стратегии за маршрутизация. Реализация на машини на състояние (крайни автомати) с помощта на актьори. | 4 |
|  | Компютърно зрение и разпознаване на обекти – запознаване с библиотеките *OpenCV-Java, OpenCV-Python* и *NumPi*. Практически примери за обработка на изображения и компютърно зрение с *Pi Camera Module v2* | 4 |
|  | Планиране на действия и манипулиране на обекти - декомпозиция на проблема и алгоритми за планиране и извод. Представяне на състоянията целите и действията. Търсене на път в пространство на състоянията - *forward* и *backward chaining*. Евристично търсене. Граф на планиране (*planning graph*). Използване на съждителна логика. *STRIPS (Stanford Research Institute Problem Solver)* планиране. Практическа реализация на езика *Java*. | 4 |
|  | Самообучение и адаптивно поведение при интелигентни агенти. Проектиране на самообучаващи се системи. Индуктивно логическо самообучение - представяне на хипотези, научаването на понятие като търсене в пространство на хипотезите, пространство на версиите (*version space*), *inductive bias*. Използване на *дърво на решенията (decision tree)*. Използване на *невронни мрежи* (*neural networks*). *Deep learning*. | 4 |
|  | *FIPA* стандарти на за изграждане на многоагентни системи. *Belief–Desire–Intention (BDI)* модел на човешките практически разсъждения и езици за комуникация между интелигентни агенти *(Agent Communication Languages – ACLs)*. Представяне на знания с помощта на онтологии и W3C стандарти за изграждане на *Семантична Мрежа (Semantic Web): Resource Description Framework (RDF)* и *Web Ontology Language (OWL)*. Практическа реализация на онтология с *Protege*. Достъп до онтологията с Java *- Apache Jena Ontology API*. | 4 |
|  | Финален тест и обсъждане на курсовите проекти. Допълнителни въпроси. Перспективи пред *IoT* и социалната роботика. | 4 |

***Конспект за изпит***

|  |  |
| --- | --- |
| **№** | **Въпрос** |
|  | „Интернет на нещата“ *(Internet of Things – IoT)*, обслужващата и социална роботика (*service and social robotics*), умни домове (*smart homes*), автономни транспортни средства (*autonomous vehicles*), интелигентни материи (*smart fabrics* *& wearable electronics*), агрегация и обработка на сложни събития (*Complex Event Processing - CEP*) в облака (*cloud computing*). Разпределена обработка на потоци от събития в реално време. Ламбда архитектура: *Apache Spark, Storm, Kafka, Apex, Flink, Beam*. Многослойна референтна архитектура на *IoT*. *Platform as a Service (PaaS)*. Наблюдение и управление на *IoT* мрежи, устройства и роботи чрез вградени и мобилни уеб/ *native* интерфейси *–* *application dashboards* с *Grafana*. |
|  | Приложение на многоагентните системи с разпределен изкуствен интелект в кооперативната, социалната и домашната роботика и автоматизация. Обектно-ориентирано, актьор-ориентирано и агентно-ориентирано софтуерно инженерство. Реактивно програмиране (*reactive programming*) и *actor model* – *Spring Reactor, RxJava, RxJS, Akka*. Отличителни характеристики на интелигентните агенти. Видове интелигентни агенти. Интелигентни агенти с хардуерна архитектура – роботика. Колаборативна и социална роботика. Платформи за изграждане на малки роботи и IoT устройства– запознаване с *Raspberry Pi 2, Lego® Mindstorms EV3* и *Arduino*. |
|  | Рационални агенти – *AIMA (Artificial Intelligence: Modern Approach)* агенти. Структура на рационални агенти. Агенти базирани на таблица. Агенти с прости рефлекси. Агенти с вътрешно състояние. Агенти базирани на цели. Агенти базирани на полезност. Агентни среди – видове и свойства. Програмна реализация на агенти и агентни среди. Примери. |
|  | Изграждане на роботи с *Lego® Mindstorms.* Програмиране на *Lego®* роботи. *Java®* програмиране на *Lego®* роботи с използване на *LeJOS – Wifi* конфигуриране, разработка с *LeJOS*, билд файлове, използване на сензори, мотори, инструменти,  *ssh* достъп до *EV3* блока, примерни *LeJOS* програми – избягване на препятствия, хващане на предмети, следване на линия. |
|  | Програмиране на *Raspberry Pi 3* роботи и IoT устройства с използване на Java библиотеката *Pi4J*. Управление на сензори и актуатори през *GPIO* интерфейс – основни протоколи за комуникация: *Serial, I2C, SPI*. Елементи на схемотехниката – аналогово-цифрови преобразуватели, *level shifters* и др. Сглобяване на малък роботс *Raspberry Pi 3.* Управление на драйвера на двигателите през *GPIO* с *Pulse Width Modulation (PWM).* Сензори за дистанция и програмиране на прости рефлекси у *Raspberry Pi* робота. |
|  | Хардуерна и софтуерна платформа с отворен код *Arduino –* хардуер, сензори и актуатори, комуникационни протоколи, езици за програмиране, среди за разработка *Arduino IDE / Eclipse (C++)*. Програмиране на *Arduino* проектисъс сензори и актуатори. |
|  | *Интернет на нещата (IoT)* - практически проекти с *ESP 8266 WiFi SOC (System-On-Chip)* модули с интегриран *TCP/IP* стек от протоколи. *IoT* архитектури и *протоколи за комуникация* (*MQTT, CoAP, AMQP*) и конфигуриране на устройства (*OMA-DM, LWM2M*). *Eclipse IoT* *платформа*, проекти: *Edje, Paho, Wakaama, Kura, OpenHAB/SmartHome, Californium, Paho, Mosquitto, Leshan, Hono, hawkBit, BIRT*. |
|  | Взаимодействие и кооперация между хора и интелигентни агенти с хардуерна архитектура (роботи). Проектиране и реализация на графични интерфейси за вградени устройства и роботи *с Java Swing /* *Java FX*  на *ARM Linux* платформа. |
|  | Разработка на уеб интерфейси за управление на *IoT* устройства с *JavaScript* библиотеките *Angular 2, TypeScript, Material Design. Angular CLI. Ionic 2* платформа за разработка на *native* мобилни приложения с *Angular* и *TypeScript*. |
|  | *Akka toolkit* за разработка на масивно конкурентни, разпределени и устойчиви към грешки събитийно-ориентирани *актьор (аctor) системи* – референции, състояние, поведение, наследници, супервайзинг стратегия, прекъсване. Наблюдение и контрол – стратегии *One-For-One* и *All-For-One*. Делегиране на задачи на наследници. Референции, пътища и адреси на актьори. Селекция по път. Обработка на изключения и грешки. Жизнен цикъл на актьор – *DeathWatch*. Изпращане и получаване на съобщения – *tell, ask, reply*. Спиране на актьори. Динамично добавяне/замяна на поведения на актьори – *become* and *unbecome* (шаблон *State*). Актьори със *stash*. Диспечери на съобщения. *Mailboxes* – типове конфигурация и приоритети. Маршрутизация на съобщения и стратегии за маршрутизация. Реализация на машини на състояние (крайни автомати) с помощта на актьори. |
|  | Компютърно зрение и разпознаване на обекти – запознаване с библиотеките *OpenCV-Java, OpenCV-Python* и *NumPi*. Практически примери за обработка на изображения и компютърно зрение с *Pi Camera Module v2.* |
|  | Планиране на действия и манипулиране на обекти - декомпозиция на проблема и алгоритми за планиране и извод. Представяне на състоянията целите и действията. Търсене на път в пространство на състоянията - *forward* и *backward chaining*. Евристично търсене. Граф на планиране (*planning graph*). Използване на съждителна логика. *STRIPS (Stanford Research Institute Problem Solver)* планиране. Практическа реализация на езика *Java*. |
|  | Самообучение и адаптивно поведение при интелигентни агенти. Проектиране на самообучаващи се системи. Индуктивно логическо самообучение - представяне на хипотези, научаването на понятие като търсене в пространство на хипотезите, пространство на версиите (*version space*), *inductive bias*. Използване на *дърво на решенията (decision tree)*. Използване на *невронни мрежи* (*neural networks*). *Deep learning*. |
|  | *FIPA* стандарти на за изграждане на многоагентни системи. *Belief–Desire–Intention (BDI)* модел на човешките практически разсъждения и езици за комуникация между интелигентни агенти *(Agent Communication Languages – ACLs)*. Представяне на знания с помощта на онтологии и W3C стандарти за изграждане на *Семантична Мрежа (Semantic Web): Resource Description Framework (RDF)* и *Web Ontology Language (OWL)*. Практическа реализация на онтология с *Protege*. Достъп до онтологията с Java *- Apache Jena Ontology API*. |

***Библиография***

***Основна:***

1. GitHub ресурси за роботика и IoT - <https://github.com/iproduct/course-social-robotics/wiki>
2. Информация за роботите ***LeJaRo*** и ***IPTPI*** – <http://robolearn.org/>
3. Сайт на *W3C* за семантична мрежа –http://www.w3.org/standards/semanticweb/
4. Официален уеб сайт на *Lego® Mindstorms* – <http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com>
5. Официален уеб сайт на ***LeJOS*** *–* <http://sourceforge.net/projects/lejos/>
6. Сайт на проекта с отворен код***Arduino*** *–*<http://arduino.cc/>
7. Ресурси на български езика за платформата ***Arduino*** *–* <http://playground.arduino.cc/Bulgarian/Nachalo>
8. ***Akka*** toolkit for concurrent & distributed applications – <http://akka.io/>
9. Сайт на *Foundation for Intelligent Physical Agents (FIPA)* – <http://www.fipa.org/>
10. Сайт на ***Spring******Reactor*** библиотека за реактивно програмиране с Java – <https://projectreactor.io/>
11. Сайт на ***RxJS*** библиотека за реактивно програмиране с JavaScript – <https://github.com/ReactiveX/rxjs>
12. Сайт на ***Akka*** платформа за разработка на actor-oriented приложения ***с*** Java/Scala – <http://akka.io/>
13. Сайт на ***Protege*** редактор за създаване на онтологии и изграждане на интелигентни системи – <http://protege.stanford.edu/>
14. Сайт на ***Apache Jena*** библиотека за достъп до онтологии и изграждане на приложения за "Семантична мрежа" с Java – <https://jena.apache.org/>
15. Сайт на ***ESP 8266 System-On-Chip (SOC)*** – <http://www.esp8266.com/wiki>
16. Murch R., Johnson T., Intelligent Software Agents, Prentice Hall, 1999
17. Russel S., Norvig P., Artificial Intelligence: a Modern Approach (3rd Ed.), Prentice Hall, 2009

**Дата: Съставил:**

20 април 2018 г ас. Траян Илиев