***Topic 8.*** Programming: Programming Paradigms and Programming Languages. OOP. Game Programming. AI (VR, AR, MR, machine learning, deep learning). Robotics.

1. General Overview of Program Development: Programming Basics. Program Planning. Program Coding. Program Testing.

2)  Programming Tools: Language Evolution. Compilers and Interpreters.

1. Programming Tools: Paradigms and Languages. Toolsets.
2. OOP
3. Game Programming.
4. AI(Artificial Intelligence) (VR, AR, MR, machine learning, deep learning).

7) Robotics.

1)

***Computer programs*** play a huge role in our life. They make life much more convenient.

Computer programming ***encompasses(охватывает)*** a broad set of activities that include

\*) ***planning****(*it is the development of a method for creating a program that will help achieve a goal.***)***

***\*) coding(***the process of creating computer programs.***)***

***\*) testing(***the process of research, testing a software product, which aims to check the ***correspondence(соответсвия)*** between the actual behavior of the program and its expected behavior on a final set of tests selected in a certain way

***\*)documenting(***explanations of what the code does, how it works.).

2)

Programming language-it is a ***sequence(последовательность)*** of commands that are built according to certain rules. When writing code, specific words, functions, and operators are used, and each element has a clear meaning that a computer understands.

***\*)low-level language***-programming language close to programming directly in ***machine codes***(a system of commands of a particular computer, which is interpreted directly by the processor or microprograms of this computer.)

\*)***high-level language***-a language that is more understandable to the programmer and does not work directly with the hardware

\*\*)

*\*)first-generation programming language*, 1GL-programming languages ​​that work at the instruction level of the processor of a particular machine.

\*)(2GL)- were created in order to ease the hard work of programming by moving in language expressions from low-level machine concepts closer to how a programmer usually thinks.

\*)(3GL)- The main ***distinguishing(отличительная)*** feature of third-generation languages ​​was hardware independence, that is, the expression of an algorithm in a form that does not depend on the specific characteristics of the machine on which it will be executed.

\*)(4GL)- These are development environments. This generation includes object-oriented languages. These languages ​​build in powerful operators that allow one line to describe the kind of functionality that would require thousands of lines of source code in younger generation languages.

\*)(5GL)- These are development environments where programs are created using visual tools that do not require deep programming knowledge.

\*\*\*)

Both the Compiler and the Interpreter have the same purpose - to convert high-level language instructions into a binary form understandable by the computer.

\*)***The compiler*** translates the high-level language into machine language.

The compiler first scans the entire program and then translates it into machine code

***\*)The interpreter***, before converting the code to machine language, it converts this code into an intermediate language. Each piece of code is interpreted and executed separately and sequentially.

3)

***A programming paradigm*** is a collection of ideas and concepts that define the style of writing computer programs.

The most popular paradigms are: event-driven, procedural, object-oriented and declarative.

***Event-driven paradigm***(Событийно-ориентированное программирование) is that users select any UI(user interface) element and then we **handle**(обрабатываем) event of its selection. (Visual basic, C#)

According to ***procedural paradigm*** we provide the computer liner steps to carry out some task.

(BASIC, Ada, Pascal, Fortran, COBOL)

In object-oriented paradigm a program consists of classes, objects and their state.

(C#, C++, Java)

Declarative paradigm is used to describe a problem.

(Prolog,SQL)

4)

Let’s speak about ***object-oriented paradigm*** is detail. As a said, in object-oriented paradigm a program consists of classes, objects and their state.  
***\*) An object*** is a unit of data which represent an abstract or real world ***entity(сущность)*** like human, place etc.   
***\*)A class*** is ***a template(шаблон)*** for objects creation.   
***\*)A class state*** consists of **attributes** and **methods**.   
 /)**A class attributes** define the characteristics of a set of objects. Each attribute has **name, date type,** and **scope(областьдействия\объем).** A scope can be defined as public or private. A public attribute means that it’s available to use from any part of a program. A private attribute means that it’s available to use only form the class in which it is defined.

**\*)Inheritance**(наследование) means the ability of one class which called **child class** or **subclass**(подкласс) to inherit any characteristics and methods, which was marked as public or protected, from other class called **superclass** or **parent class.**

\*)**A method** is a segment of code that defines an action.

A message is a method activator.

\*\*)**Polymorphism**, sometimes called «**overloading**», is the ability to redefine a method in a subclass. It allows to **simplify**(упростить) program code.

***5)***

***Game programming*** is the branch of software engineering related to creating video games.

\*)Game programmers are translating **the project's vision(видение проекта)** into code to create a playable game.

To become a game programmer, you should know any programming language like C++, C#, Java etc.

Also you should have the experience with **engines(типо двигатель, но движок**) like Unity, Unreal engine etc.

And also you should have strong math and physics base. Game industry is very **in demand**(**востребована)** and it is a good time to get into it. But I think it’s very difficult to do.

There is wide list of game programmer’s types like UI programmers, AI (Artificial Intelligence)programmers, Graphics programmers etc.

\*)Let’s speak about development process of game.

**The first** process is prototyping. During this process programmers produce prototypes of gameplay ideas and features.

**The second** process is game design.

**The third** process is production. During this process developers write a code for a game.

**The next** process is testing. Here testers test game’s correctness and stability.

The game development end with **maintenance.**

6)

Artificial Intelligence is basically the mechanism   
**to incorporate(включения)** human intelligence into machines through/ a set of rules(algorithm).

**\*)Machine Learning(ML)** is basically the study process which provides the system(computer) to learn automatically on its own through experiences it had. **ML is an application or subset(подмножество) of AI.**

**\*)Deep Learning(DL)** is basically a **sub-part(часть)** of the broader family of Machine Learning which uses **Neural Networks** to **mimic(имитация)** human brain-like behavior.

\*)**Virtual Reality (VR)** is a digital simulation created with virtual reality **helmets(шлемы).**

Using AR(**Augmented Reality(дополненная)**), we are watching on the world not directly, but via some devices which **embed(встраивать)** some objects into real world. This is the main difference between AR and VR, AR takes as a basis the real world and **Augments**(дополняет) some virtual object into it.

7)

Robotics **involves** developing devices to perform tasks which require high **precision(точность)** or dangerous for humans. The brain in an advanced industrial robot today works at about ten million instructions per second but to achieve human’s intelligence they must to achieve about 100 trillion operations per second. To change **mOtions(движения)**, robots use sensors, which can detect light, sound, touch and **heat(тепло)**.

Robots can be involved in many spheres like industry, military etc.

Androids are **anthropomorphic(перенесение человеческого образа и его свойств на неодушевлённые предметы и животных)** robots designed to look and behave like a human. They can walk, talk and understand human speech.