**Вагнер А.А. 5130901/20003**

**Linkers: 10, Vocabulary: 23, Total: 608**

**Spring 2023-2024 Term 4**

**ИКНТ/ИКиЗИ/ФизМех**

**Monologue on *PROGRAM DESIGN AND COMPUTER LANGUAGES (UNIT 4)***

|  |  |
| --- | --- |
|  |  |
| **Step 1. Introduction**  1. Start with a hook sentence that will attract the listener’s attention, a quote, a proverb, etc.  2. Lead your speech steadily to the main part of your talk.  3. The introduction may consist of 3-6 sentences. | Program design is the art of arranging code in a way that both humans and machines can understand. As the great computer scientist Alan Perlis once said, 'A language that doesn't affect the way you think about programming is not worth knowing.' In this monologue, we will delve into the intricate world of program design and explore how different computer languages shape our approach to problem-solving. |
| **Step 2. Important traits of a programming language**  2.1. What are the traits important for constituting a programming language?  2.2. What are the contexts in which programming languages are used? | A programming language is a construct designed for the purpose of expressing algorithms and instructions in a way that can be understood by computers. The expressive power of programming languages varies, with some being Turing complete, meaning they can perform any computation that can be described algorithmically.  Programming languages are defined by formal grammars, which outline the rules for forming valid statements in the language. In the context of programming, languages are classified based on the computations they are capable of expressing. Different languages are suited for different tasks, and choosing the right language for a particular project can greatly impact its success. However, there are some skeleton key languages that suit almost any task. |
| **Step 3. Program design**  3.1. How are programming languages different from natural languages?  3.2. Speak about the trends in the development of programming languages | Programming languages are distinct from natural languages in several key ways. To be exact, they offer a greater degree of precision and completeness, allowing for complex instructions to be communicated effectively. Programs written in programming languages can be executed in a batch process without human interaction, automating tasks efficiently. These languages can also be altered to meet new needs, combined with other languages, or fall into disuse as technology evolves.  The trends in the development of programming languages reflect the diversity of contexts in which they are used and the range in expertise required to work with them. Languages undergo modification to solve problems using a higher level of abstraction, making them more efficient and versatile. Additionally, programming languages are often tied to the underlying hardware, optimizing performance and functionality. As technology advances, new languages emerge while older ones become obsolete, highlighting the dynamic nature of the field. |
| **Step 4. History of programming languages**  4.1. The predecessors of programming languages. What were they?  4.2. Speak about the generations of programming languages. | The evolution of programming languages can be traced back to the early days of computing when machine code was used to encode data and direct mechanical processing on electrically powered digital computers. As the need for more efficient and user-friendly programming grew, the first-generation programming languages emerged, characterized by low-level languages that closely mirrored the computer's hardware architecture.  With the development of (relatively) high-level programming languages, such as FORTRAN and COBOL, programmers were able to use forms resembling English to write code, therefore making it easier to understand and maintain.  As technology advanced, second-generation programming languages were introduced, incorporating a formal system in mathematical logic to extract the underlying essence of mathematical concepts, removing any dependence on the computer's hardware. I, personally, find last gen programming languages to lack some kind of challenge of usage, due to having long ago created solutions for basically any problem one can face during programming session. |
| **Step 5. CREATIVE THINKING**  Introduce your own extra idea(s) on program design and computer languages that hasn’t/haven’t been mentioned before. Substantiate your choice. | One idea that hasn't been mentioned yet is the concept of domain-specific languages (DSLs). DSLs are specialized computer languages tailored to a specific domain or problem space, allowing for more concise and expressive code that closely aligns with the domain's concepts and requirements. By using DSLs, developers can write code that is easier to understand and maintain, leading to improved productivity and code quality. |
| **Step 6. Conclusion**  Summarise the ideas of steps 2,3,4,5. | In conclusion, the world of programming languages is vast and ever-evolving, with each language offering unique features and capabilities that shape the way developers approach problem-solving. From low-level machine code to high-level domain-specific languages, the evolution of programming languages has greatly influenced the efficiency and effectiveness of software development and undoubtedly greatly changed the world we live in nowadays allowing use of vast and highly efficient systems, for example, Internet or unmanned means of transportation. |