NATIONAL RESEARCH UNIVERSITY

HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science

Bachelor’s Programme “Applied Mathematics and Informatics”

**Software Project Report on the Topic:**

**Development of Materials for Teaching Programming in Python**



**Fulfilled by:**

| Student of the Group  БПАД212  Kulakov Denis Alexeevich | ————————  *(signature)* | May 31st, 2024  *(date)* |
| --- | --- | --- |

**Checked by the Project Supervisor:**

Voznesenskaya Tamara Vasilievna

Associate Professor

Big Data and Information Retrieval School, Faculty of CS, HSE University

| ——————————  *(signature)* |  | June 2nd, 2024  *(date)* |
| --- | --- | --- |

**Moscow 2024**

# Table of contents

[**Table of contents 2**](#_ctn5f0ke4ne9)

[**Abstract 3**](#_7ng70nyp63wa)

[**Introduction 4**](#_keacdjhs7a73)

[Subject area 6](#_ulyk0sb1shyh)

[Relevance 6](#_7yzpbqignxok)

[Main goal 9](#_haxvu1nc2aqe)

[Main tasks 9](#_9ffznhwkgg42)

[Instruments 9](#_u4ghjmw1qtv5)

[**Basic terms and definitions 10**](#_y6jxpvxp60tb)

[**Project implementation 12**](#_8v2hqjafimzw)

[**Existing alternatives and analogues: comparative analysis 19**](#_axoz671ivip8)

[**Results and conclusion 20**](#_9i22mknvd2lt)

[Results description 20](#_iae7jgpnwd1z)

[Perspectives for future work 20](#_56p7j092civu)

[Conclusion 21](#_a2a7zjgz8gfo)

[**Sources 22**](#_bb7htyjv7m8k)

# Abstract

This project seeks to address the growing need for effective computer science education at the secondary school level. This project focuses on creating a comprehensive set of instructional materials tailored for teaching Python, a language renowned for its readability and versatility. By incorporating modern software development tools like Docker and Git, the project ensures that students are not only learning programming fundamentals but also gaining practical knowledge of current industry practices. Docker is introduced to teach students about containerization, providing hands-on experience with creating consistent development environments. Git is used to familiarize students with version control and collaborative workflows, essential skills in today's software industry.

The instructional materials developed include slide decks, code examples, and self-evaluation tools, all designed to facilitate an interactive and engaging learning experience. The project emphasizes practical applications of Python, demonstrating its use in various real-world scenarios to inspire and motivate students. Additionally, a comparative analysis of existing educational programs highlights the strong sights of this project, particularly its focus on integrating modern development practices.

# 

# Introduction

The project titled “Development of Materials for Teaching Programming in Python” seeks to develop a set of pedagogical materials for teachers to use to teach Python programming to the secondary school audience. Python is selected as the introductory language in computer science education with increasing frequency and for good reason. Its syntactic clarity and flexibility make it an excellent tool both for the teaching of foundational programming concepts and for building complex applications. The goal of this project is to carefully construct a set of instructional materials that is both comprehensive and pedagogically rigorous, in line with the academic needs of modern students.

This effort is underpinned by the use of modern software development tools, Docker and Git, that serve to expose students to current industry standards and practices. Docker is a tool for crafting lightweight, self-contained computing environments that remove disparities between computational settings that might arise from differential local development environments. Git is a distributed version control system that exposes students to workflows of collaborative and iterative development that are in widespread use throughout the software industry today. The project aspires to deliver advanced educational experiences through the construction of master classes. These are designed to provide exposure to expert knowledge and sophisticated programming concepts, and will build on the foundational curriculum to enrich the learning experiences of the students.

The ultimate goal is to construct an educational guide that not only transmits technical knowledge, but also equips students with the skills they will need to manage the intricacies of modern computational problem solving. By addressing this goal in a systematic, evidence-based fashion, the hope is that the project will make a meaningful contribution both to the broader field of computer science education and the pipeline of future technologists who will enter the technology workforce.

## 

## Subject area

The subject area of this project is the field of computer science education with a special focus on:

1. Modern software development tools:
   1. **Docker**: Students are introduced to containerization concepts, learning how to create consistent development environments using Docker. This exposure helps them understand modern software deployment practices.
   2. **Git**: Version control is taught using Git, enabling students to learn about collaborative workflows and the importance of version management in software development.
2. Real-world applications:
   1. The project emphasizes the practical applications of Python, demonstrating its use in various real-world scenarios. This approach aims to inspire students and show the relevance of programming skills in diverse fields.

## Relevance

The project is highly relevant in today’s educational and technological landscape for several reasons:

1. **Wide application**
   1. Python is used in a wide range of applications, from web development and automation to data analysis and machine learning. This makes it a valuable language for students to learn, as it can be applied in many different contexts.
   2. Its popularity among professionals and educators ensures a strong community support and a wealth of resources available for learners.
2. **Industry demand**
   1. There is an increasing demand for programming skills across various industries, including technology, finance, healthcare, and more. Python, in particular, is widely used due to its versatility and ease of learning.
   2. Proficiency in Python opens up numerous career opportunities.
3. **Educational needs**
   1. Schools and educational institutions are increasingly incorporating programming into their curricula to prepare students for the future job market.
   2. Python’s syntax is designed to be intuitive and easy to read, which helps beginners quickly grasp fundamental programming concepts. This lowers the barrier to entry for new learners and makes programming more accessible.
4. **Integration of modern development practices**
   1. Use of **Docker**: Introducing students to Docker helps them understand the importance of containerization in modern software development. It provides practical experience with creating consistent development environments, which is crucial for real-world applications. Knowledge of Docker is increasingly becoming a valuable skill in the job market, especially for roles in DevOps and cloud computing.
   2. Use of **Git**: Teaching Git exposes students to version control systems and collaborative workflows, which are essential skills in the software industry. Understanding Git helps students manage code changes effectively and collaborate with others on projects. Familiarity with Git prepares students for industry-standard development practices, making them more competitive in the job market.

## 

## 

## Main goal

The goal of this project is to equip school students with basic information in programming on Python and essential software development skills.

## Main tasks

1. Study the materials provided by the supervisor
2. Study already-existing accessible analogues and alternatives
3. Develop the curriculum and the idea behind the desired course
4. Create the materials
5. Prepare the report on the project
6. Display the results and the products of the project.

## Instruments

1. Google Drive
2. Google Docs
3. Jupyter Notebook
4. Google Slides
5. Git
6. Docker
7. Google Collab
8. SlidesGo

## 

# Basic terms and definitions

* **Programming language** - an artificial language invented to concretise and record computations that can be performed by a machine, particularly a computer.
* **Python** - a high-level, interpreted programming language known for its readability and broad applicability in various programming tasks.
* **Educational materials** - resources created to support learning objectives, including texts, videos, and/or interactive exercises.
* **Git** - a distributed version control system used to control source code by programmers collaboratively developing software.
* **Repository (repo)** - a storage location where the project's files and history are stored.
* **Commit** - a snapshot of a project at a specific point in time (saving the project's current state).
* **Branch** - a separate line of development.
* **Merge** - the process of combining changes from different branches.
* **Clone** - creating a copy of an existing repository.
* **Remote** - a version of your repository that is hosted on the internet or network.
* **Staging area** - a place where changes are gathered before committing.
* **Pull request (merge request)** - request to merge code from one branch into another, usually from the developer’s personal branch into a main or development branch.
* **Rebasing** - a process of moving or combining a sequence of commits to a new base commit. It is commonly used to keep a feature branch up to date with the main branch.
* **Cherry-picking** - the act of choosing a commit from one branch and applying it to another.
* **Stashing** - allows one to save your changes temporarily without committing them.
* **Docker** - a set of platform-as-a-service products that use OS-level virtualization to deliver software in packages called containers.
* **Image** - a lightweight, stand-alone, executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and dependencies.
* **Container** - a runtime instance of an image.
* **Dockerfile** - a text file that contains a list of commands to build a Docker image.
* **Docker hub** - a cloud-based repository where Docker users and partners create, test, store, and distribute container images.

# Project implementation

# 

The project includes several key branches and steps of tasks aimed at creating a comprehensive and effective educational program for secondary school students. The list of these tasks may be found below:

1. **Master classes for school students:** 
   1. Сurriculum development: identify key Python programming concepts suitable for master class treatment.



* 1. Resource creation: produce supplementary materials, such as slide decks and code examples.



* 1. Develop self-evaluation tools to assess student understanding and the effectiveness of the master classes.





1. **Containerization concepts with Docker:**
   1. Introduce the concept of containers, explaining their advantages



* 1. Develop tutorials that demonstrate the setup and management of Docker containers.





* 1. Design exercises that guide students through the process of building, running, and managing Docker containers.



* 1. Align Docker learning modules with the existing Python curriculum to reinforce the application of containerization in software development.



1. **Working with Git:**
   1. Create materials that cover basic Git commands and workflows, including cloning repositories, committing changes, and pushing to remote repositories.



* 1. Teach branching strategies and how to resolve merge conflicts effectively.



* 1. Design exercises that help students to study materials about collaboration work using version control systems like Git.



# Existing alternatives and analogues: comparative analysis

In order to understand the positioning and potential impact of the project, it is important to analyze existing similar educational programs in Russia. This comparative analysis will highlight the strengths and limitations of current offerings and identify areas where this project can provide additional value.

**Yandex.Lyceum** is a popular educational program designed by Yandex, one of Russia's leading technology companies. It offers a two-year course on programming for high school students, focusing primarily on Python.

**Kaspersky CyberSecurity for the Next Generation** initiative focuses on educating students about cybersecurity, including basic programming skills necessary for understanding cybersecurity concepts.

**The Moscow Coding School** offers various programming courses for different age groups, including Python programming for high school students.

**Codewards** is an online platform offering coding courses for school students, with a strong emphasis on game-based learning.

However, all the above-mentioned platforms show little focus on integration of modern tools while paying much attention to the theoretical part of programming. Thus, unlike many existing programs, this project will incorporate Docker and Git, providing students with practical experience in modern software development environments and version control systems.

Also, the educational materials will be designed to be flexible, allowing adaptation to different teaching styles and educational needs. This contrasts with some existing programs that have more rigid, predetermined curricula.

# Results and conclusion

## 

## Results description

The results of the project is the creation of a 4-week-long Git course, a 2-week-long Docker course and a workshop with general information about the Python programming language for secondary school students. The materials are placed on a publicly-shared google folder:

## Perspectives for future work

For this project several perspectives for future work can be identified:

1. Create an online platform that hosts educational materials, interactive exercises, and tutorials. This platform can provide a more accessible and flexible learning experience for students.
2. Develop training programs and workshops for teachers to help them effectively use the new curriculum and materials in their classrooms.
3. Establish a community platform where students and teachers can share their experiences, collaborate on projects, and provide feedback on the curriculum.

## Conclusion

The project "Development of Materials for Teaching Programming in Python" successfully achieves its aim of creating comprehensive and effective educational resources dedicated to secondary school students. By focusing on Python, a language that is both powerful and accessible, the project addresses the essential need for programming education in today's technology-driven world.

Throughout the project, the integration of modern development tools like Docker and Git was paid significant attention. Docker's ability to create consistent development settings and Git's collaborative version control workflows ensure that students gain valuable skills that are highly relevant in the job market.

The instructional materials developed have been designed to be engaging and adaptable to various educational contexts. The emphasis on practical applications of Python, coupled with self-evaluation tools, helps students understand the relevance of their learning and track their progress effectively.

Comparative analysis with existing programs underscores the contributions of this project, particularly its focus on integrating modern tools and providing flexible learning paths. This approach not only fills gaps in current offerings but also sets a new standard for computer science education at the secondary school level.

In conclusion, this project makes a meaningful contribution to computer science education. By preparing students for the complexities of contemporary computational problem-solving, it helps build a pipeline of future technologists ready to meet the challenges of the evolving tech landscape.

# Sources

1. Python Software Foundation. Python Programming Language. Retrieved from <https://www.python.org/>.
2. Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239), 2.
3. Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.
4. Yandex. Yandex.Lyceum. Retrieved from <https://lyceum.yandex.ru/>.
5. Kaspersky Lab. CyberSecurity for the Next Generation. Retrieved from <https://www.kaspersky.com/>.
6. Moscow Coding School. Programming Courses. Retrieved from <https://codingschool.moscow/>.
7. Codewards. Game-based Learning Platform. Retrieved from <https://www.codewards.com/>.
8. GitHub. The Git Distributed Version Control System. Retrieved from <https://github.com/>.
9. SlidesGo. Free Google Slides and PowerPoint Templates. Retrieved from <https://slidesgo.com/>.