Co-funded by the Erasmus+ Programme of the European Union

HerTour4Youth

2020-1-UK01-KA227-YOU-094543

IO1- A3: Компендиум за дигитален дизайн
Модул 4 Компютърно програмиране

КД2 – Сътрудничество за иновации и обмен на добри практики
Творчески партньорства

KA2 - Cooperation for innovation and the exchange of good practices
Partnerships for creativity

Подкрепата на Европейската комисия за изготвянето на тази публикация не представлява одобрение на съдържанието, което отразява възгледите само на нейните автори, и Комисията не носи отговорност за каквото и да е използване на информацията, съдържаща се в нея.

Версия:	Дата	Автор	Описание	Действие	Страници
1.0	15/10/2021	CIVIC	Създаване	C	12

ИСТОРИЯ НА ДОКУМЕНТА

(*) Действие: C = Създаване, I = Добавяне, U = Актуализиране, R = Заместване, D = Изтриване

ЦИТИРАНИ ДОКУМЕНТИ

ID	Референтен номер:		Заглавие
1	2020-1-UK01-KA227-YOU-094543		Проект HerTour4Youth
2			

ПРИЛОЖИМИ ДОКУМЕНТИ

ID	Референтен номер:		Заглавие
1	Deliverable IO1.A2		Разработване на методика за обучение
2			

Съдържание

1. Въведение

Ако се чудите какво е компютърно програмиране, просто си представете много проста компютърна машина в къщата си (тя може да бъде термостат, вашия рутер за връзка и т.н.). Сега, просто знайте, че този компютър не работи сам, но е програмиран от някого. Че някой е компютърен програмист. Всеки компютър се нуждае от набор от инструкции, за да функционира гладко. Точно за това е компютърното програмиране: то може да се състои от прост набор от инструкции за улесняване на конкретни действия до сложен набор от инструкции, включващи четене и сортиране на данни.

1.1 Резултати от обучението

След завършването на този модул ще можете да:

- **Имате основно разбиране на принципите на програмирането**
- **Имате по-задълбочени познания за това как работят ИКТ инструментите**
- **Прилагате програмни принципи**

1.2 Ключови думи

- **ИКТ**
- **Компютърен програмист**
- **Езици за програмиране**

1.3 Прогнозно времетраене

1 час

1.4 Речник на термините

ИКТ (ICT): означава „Информационни и комуникационни технологии“ ИКТ се отнася до технологии, които предоставят достъп до информация чрез телекомуникации. (TechTerms, 2021)

Компютърен програмист: компютърен програмист е лицето, отговорно за създаването на инструкции за компютър, които да изпълни чрез писане и тестване на код, който дава възможност на приложения и софтуерни програми да работят успешно. (Техопедия, 2014 г.)

Езици за програмиране: езиците за компютърно програмиране са някой от различните езици за изразяване на набор от подробни инструкции за цифров компютър. (Д. Хемендингер, 2021 г.)

Аналогов: Прилагателно, което се отнася до механизъм или устройство, в което информацията е представена от непрекъснато променливи физически количества. Това е противоположното на дигитален. (Мериам Уебстър, 2021)

Изкуствен интелект Изкуственият интелект кара компютри и машини да имитират възможностите за решаване на проблеми и вземане на решения на човешкия ум, без да се ограничава до методи, които са биологично наблюдаеми." (IBM, 2020)

Машинно обучение: Машинното обучение е клон на Изкуствения интелект (ИИ) и компютърните науки, който се фокусира върху използването на данни и алгоритми, за да имитира начина, по който хората учат, като постепенно подобрява точността си. (IBM, 2020 Г.)

Бинарен (двоичен): Бинарна е базова 2-числена система, измислена от Готфилд Лайбниц, която е съставена само от две числа или цифри: 0 (нула) и 1 (едно). Тази система за номериране е основата за всички двоичен код, който се

използва за писане на цифрови данни като инструкциите на компютърния процесор, използвани всеки ден. (ComputerHope, 2021)

2. Компютърно програмиране

2.1 Въведение

В нашия воден от компютри свят знаем, че компютрите могат да правят много забележителни неща.
Бързият технологичен напредък, на който станахме свидетели през последните няколко десетилетия, има широко обхватен ефект върху това как работи всичко. Трябва да погледнете под повърхността, за да видите къде са се случили най-значимите промени. Компютрите замениха милиони работни часове и складове на аналогови машини с по-бързи, по-безопасни и по-надеждни системи.
Тъй като компютрите работят по код, се вижда защо можете да го намерите навсякъде. Компютрите ще продължат да заменят остаряла технология във всичко – от микровълновите печки до електроцентралите. И наличието на код в нашето ежедневие ще се увеличава.

Също така почти всеки в днешно време притежава лаптоп, който може да се използва за прости компютърни процеси, като създаване на електронна таблица или създаване на прости документи. Има обаче и хора, които използват лаптопа си по по-сложен и професионален начин. Това може да е за извършване на милиони финансови сделки на ден и контрол на инфраструктурата, която прави възможен съвременния живот.
Казвайки това е важно да запомните, че никой компютър не може да направи нищо, ако не е програмиран от човешко същество. В този случай говорим за фигурата на компютърния програмист.

Компютърният програмист е професионалната фигура, отговорна за писането на кодове, или *кодирането*, които инструктират компютъра, приложението или софтуерната програма за това как да работи.

Както бе споменато по-рано този набор от инструкции може да бъде съвсем прост и целият процес на кодиране може просто да изисква добавяне на две числа, но също така би могъл да включва по-сложни неща като четене на данни от температурни сензори за регулиране на термостат, сортиране на данни, за да завърши сложно планиране или критични доклади или превеждане на играчи чрез многопластови светове и предизвикателства в игрите.

Докато компютърните науки се занимават с теоретични идеи на високо ниво, почти всеки аспект на съвременния живот разчита на кодирането. Всяко приложение на нашия телефон, таблет или компютър използва компютърни езици за да работи.

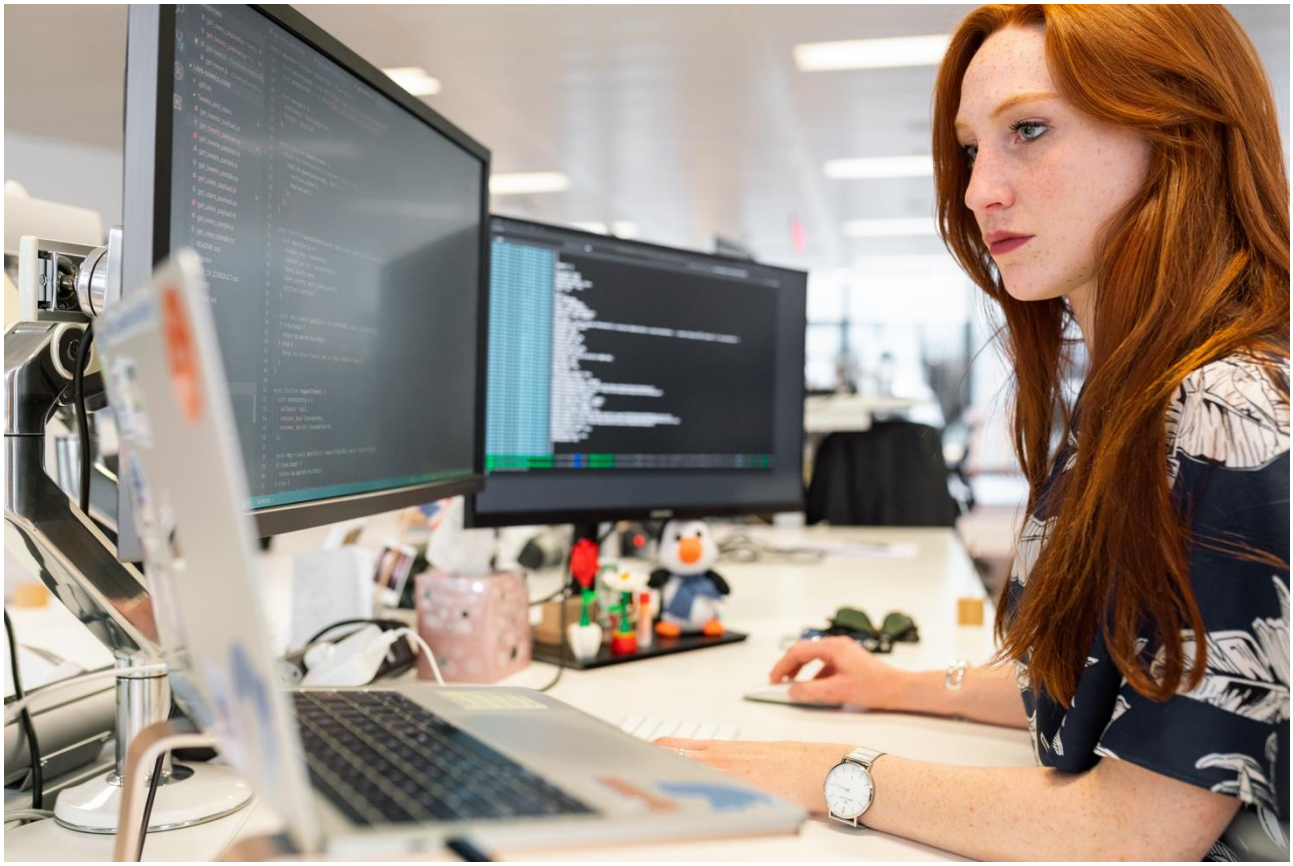
Източник: https://pixabay.com/photos/code-coding-computer-data-1839406/

Други цифрови системи като смарт телевизори и калкулатори го използват също. На практика всеки нов автомобил го използва, за да контролира всичко – от климатичните системи до инжекторите за гориво. Градовете ползват компютри за регулиране на сигналите на светофарите.

Системите, които преди бяха аналогови, сега се рационализират с помощта на компютъризирани системи. Това позволява на инженерите да изградят по-ефективна и не толкова скъпа система, структура и машина. Освен това някои от най-модерните технически области (като изкуствения интелект и машинното обучение) използват кодиране.[1]

2.2 Компютърен програмист

[1] Weinstein, J. (2021, January 24). What Is Coding? Coding Definition and Uses. Career Karma. https://careerkarma.com/blog/what-is-coding-used-for/



Източник: https://unsplash.com/photos/64YrPKiguAE

Както бе споменато преди компютърният програмист е лицето, отговорно за създаването на инструкции за компютъра, които да изпълни чрез писане и тестване на код, който дава възможност на приложения и софтуерни програми да работят успешно.

Компютърният програмист може да бъде наречe и програмист, кодер, разработчик или софтуерен инженер. Също така терминът често се използва за обозначаване на самостоятелен разработчик на софтуер, разработчик на мобилни приложения, Уеб разработчик, софтуерен анализатор, вграден разработчик на фърмуер и така нататък.

Компютърният програмист е квалифициран професионалист, който кодира, тества, отстранява грешки и поддържа изчерпателните инструкции, известни като компютърни програми, които компютрите трябва да следват, за да изпълняват специфичните си функции[2].

Някои от често срещаните задачи, които един компютърен програмист трябва да овладее, включват:

- **Тестване на производителността на софтуера.**
- **Разрешаване на проблеми с компютърния софтуер.**
- **Модифициране на софтуерни програми за подобряване на производителността.**
- **Писане на код за компютърно програмиране.**
- **Сътрудничество с други хора за разрешаване на проблеми с информационните технологии.**

Обичайно е компютърните програмисти да се класифицират в два вида: програмисти на системи и програмисти на приложения.

И двете роли използват кодиране. Разликата между двете е, че програмистите на приложения извършват кодиране за управление на определена задача, като например кодиране на програма за наблюдение на запасите в рамките на дадена фирма.

От друга страна програмистите на системи кодират програми за поддръжка и контрол на системния софтуер, включително системи за управление на бази данни и операционни системи (OSs).

[2] Techopedia. (2014, July 30). Computer Programmer. Techopedia.Com. https://www.techopedia.com/definition/6589/computer-programmer

2.1.1 Езици за програмиране

Също като езиците използвани от човека, компютрите имат и свои езици. Наистина съществуват множество езици за компютърно програмиране, които програмистите да използват за комуникация с компютъра, за изпълнение на набор от конкретни задачи.

Частта от езика, която един компютър може да разбере, се нарича "бинарна/двоична". Преводът на програмен език в бинарен е известен като "компилиране". Всеки език за програмиране има свои обособени черти. Понякога обаче между тях[3]има прилики.

Някои от най-често срещаните езици за компютърно програмиране са:

- **C**
- **SQL**
- **PHYTON**
- **JAVA**
- **R**
- **HTML[4]**

[3] Writers, S. (2021, September 27). Guide to Programming Languages | ComputerScience.org. Get an Education the World Needs | ComputerScience.Org. https://www.computerscience.org/resources/computer-programming-languages/

[4] Gallagher, J. (2021, May 4). The Most Popular Programming Languages. Career Karma. https://careerkarma.com/blog/top-programming-languages-2021/

Source: https://pixabay.com/illustrations/programming-languages-icon-898961/

Много езици за програмиране са сравнително прости, но те правят различни неща. Например, един от най-популярните езици, JavaScript, се използва предимно за уеб страници и front-end разработка. От друга страна, Python се използва както за цялостни софтуерни програми, така и за уебсайтове.

Който и софтуер да избере разработчика, обикновено е по негова преценка, тъй като често повече от един език за програмиране може да става за една и съща цел.

Познаването на един от повечето от тези програмни езици е от съществено значение, ако искаме да постигнем кариера не само като компютърен програмист, но и като софтуерен инженер или всяка работа в областта на науката за данни, която изисква високи технологични познания.

2.1.2 Процес на програмиране

Основната цел на програмирането е като цяло да се създаде решение за даден проблем. Проблемът може да бъде нещо голямо в международен мащаб или просто нещо малко, да ни освободи от скуката.

Източник: https://pixabay.com/illustrations/programming-languages-icon-898961/

Разработването на програма включва стъпки, подобни на всяка задача за решаване на проблеми. В процеса на програмиране има четири основни стъпки:

- **Определяне на проблема: на този етап се опитвате да идентифицирате проблема. Въпреки това би било разумно да се каже, че това, което наистина се опитвате да идентифицирате, е решението на проблема, тъй като се опитвате да дефинирате това, което искате да постигнете.**

Първото нещо, което трябва да направите, е да анализирате изискванията. Наистина, трябва да разберете какво ще се изисква да направи вашата програма.

Второто нещо, което трябва да направите е да спазвате списъка с изисквания и да решите какво точно трябва да направи вашето решение, за да ги изпълни. Понякога един проблем може да изисква няколко решения, така че е важно да разберете кое е най-доброто решение за всеки проблем.

По този начин се опитвате да посочите още веднъж какво решение трябва да приемете за проблема си.

- **Планиране и проектиране на решението: това като цяло е най-трудната задача от всички, защото изисква да разберете как ще превърнете тези спецификации и изисквания в работна и функционална програма.**

  Проектирането на решение означава да се опишете на по-високо ниво всички стъпки, които компютърът трябва да следва, за да функционира правилно или да изпълни конкретна задача.

  Тази задача е най-трудната по различни причини: например може да се наложи да научите малко повече за функциите на компютъра и избраните от вас езици за програмиране, за да видите какви неща могат да бъдат лесни или трудни; също както е споменато по-рано, даден проблем да може да изисква различни решения и след като бъде идентифицирано решение, е важно да анализирате всичките му силни и слаби страни и да направите правилния избор.

  Въпреки това, след като трудната част свърши, трябва да бъде завършена частта на проектирането. Това означава, че имате много ясна представа как компютърът ще отговори на спецификацията ви, ще спазва вашите изисквания и най-накрая ще реши първоначалния ви проблем.

- **Кодиране на програмата: В тази задача ще трябва да инструктирате компютъра си как да извърши проектираното от вас. Тази фаза включва 3 етапа:**
  - **Кодиране: Кодирането означава да преведете проекта си във функционална програма, като използвате език за програмиране. Този етап на практика включва да седнете и да започнете да въвеждате кодове на компютъра. Точно като есе, кодирането на програма изисква да включите неща като заглавия, страница със съдържание, въведение и препратки. След като приключите с кодирането си трябва да го подадете на компютъра, за да видите как реагира компютърът и какво прави от него.**
  - **Компилиране: Фазата на компилиране (съставяне) изисква да превърнете писмената програма в език, който компютърът разбира, тоест на бинарен/двоичен машинен код. Има някои програми, наречени компилатори, които могат да направят това за вас. Трябва обаче да сте много прецизни, когато пишете програмата си в компилатор, защото той ще открие всяка минимална грешка.**
  - **Отстраняване на грешки: както бе споменато по-рано, компилаторът ще открие всяка една грешка, направена в програмата. Не се тревожи, това е много често срещана практика. Затова този етап е поставен тук. След като идентифицирате всички грешки (bugs) във вашата програма, благодарение на помощта на компилаторите, погледнете отново оригиналната програма, идентифицирайте грешките, коригирайте кода и го компилирайте отново. Този цикъл на код -> компилиране -> отстраняване на грешки (debug) често ще се повтаря много пъти, преди**

компилаторът да е доволен от него и най-накрая да имате програма, която работи. Това може да отнеме известно време.

Можете също така да решите просто да напишете малки части от вашата програма, да ги компилирате и да отстраните грешките, вместо да се фокусирате върху цялата програма наведнъж.

- **Тестване на програмата: Крайната част от процеса на програмиране включва тестване на творението ви и проверка дали работи точно по начина, по който сте искали. Знам, че може да си помислите, че компилаторът вече ви е дал правилната програма, обаче това не означава, че правилният код може да реши първоначалният ви проблем. Остава ви вие да проверите това.**

  Ако се открият възможни грешки единствената опция е вие да преразгледате програмата, коригирате грешката, като промените кода и го компилирате повторно. Не забравяйте да бъдете внимателни, когато променяте нещо в кода, защото промяната на малко нещо може да повлияе на други неща в цялата програма. Тази фаза може да се сравни с друга фаза на отстраняване на грешки.

  Накрая, след като компилирате отново, поправите грешките, отстраните грешките и тествате и след като се уверите, че програмата ви работи според вашите изисквания и спецификации, ще имате решение за проблема си![5]

2.1.3 Принципи на програмиране

[5] The Programming Process. (2020). Cs.Bham.Ac.Uk. https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html

В този момент знаем, че една от основните характеристики на компютърното програмиране е да пишете кодове. Независимо дали по принцип работите самостоятелно или в рамките на екип, не забравяйте, че кодовете ви трябва да са лесни за четене и да се поддържат за други хора.

Ето защо някои основни програмни принципи, които всеки програмист трябва да знае е да пише лесни кодове, с чисти променливи и които могат да останат силни дори след тестване и всякакви модификации.

Някои от тези принципи са:

- **KISS: Направи го простичко, глупако: Принципът KISS е един от най-важните принципи, по които да живеете в програмния свят. Означава, че трябва да пишете код възможно най-просто. Не е нужно да се опитвате и да се фукате със сложен код. Ако можете да го напишете на един ред, използвайте един ред. И не забравяйте, че някои кодове трябва да бъдат преразгледани след месеци, така че имайте това предвид и го направете просто.**
- **DRY: Не се повтаряй: Този принцип ви помага да запомните една много често срещана грешка в кодирането, тоест повторенията. Избягвайте всяко дублиране на данни и логика.**
- **Отворен/Затворен: Този принцип се грижи да помните да поддържате кодовете си отворени за разширения, но затворени за модификации. Това правило е по-удобно, когато предоставяте библиотека от услуги, отворена за използване от други. Казано накратко, ако някой друг модифицира кода ви, кодът ще се счупи или нещо друго в кода ще е засегнато и няма да работи, както обикновено трябва. Ето защо трябва да публикувате само кодове, които предотвратяват директната модификация и насърчават разширенията**

- **Единна отговорност: Този принцип гласи, че всеки клас или модул в дадена програма трябва да осигурява само една конкретна функционалност. Ето защо по този начин кодът се поддържа прост и също така избягва по-сложен процес на отстраняване на грешки; На второ място, става по-трудно да се създаде допълнителна функционалност за конкретен модул.**
- **YAGNI: Няма да ти трябва: Този принцип отстоява, че никога не трябва да опитвате и пишете кодове за функционалност, от която може да се нуждаете в бъдеще. Това би означавало, че се опитвате да решите проблем, който не съществува.**

2.1.4 Как можете да станете компютърен програмист?

Да бъдеш компютърен програмист може да ти помогне да намериш работа в индустрията за проектиране на компютърни системи и свързани с тях услуги. Много области на компютърните науки обаче изискват фигурата на компютърен програмист.

Те включват например:

- **Софтуерно инженерство**
- **Разработчик на софтуер**
- **Компютърно инженерство**
- **Компютърна графика**
- **Изкуствен интелект**

По принцип се изисква компютърните програмисти да имат университетска степен по данни и компютърни науки, което може да ви даде основите за

началото на кариерата ви. Много програмисти обаче могат да бъдат и самоуки ентусиасти, изпълнени с постоянен интерес към програмирането.

Независимо дали решавате да започнете кариерата си с университетска степен или като самоук ентусиаст, важното е да подобрите знанията си по програмните езици, да се интересувате от компютърното програмиране и също така да продължите да бъдете информирани и никога да не спирате да учите, защото компютърното програмиране е постоянна актуализираща се дисциплина.

Източник: https://pixabay.com/photos/scrabble-board-game-game-4370255/

Друго важно нещо, което много позиции за работа с компютърно програмиране може да изискват, са професионалните и нестопански сертификати, които да са ви налични.

Някои от тях включват:

- **CISCO – Сертифициран мрежов сътрудник, Сертифицирана мрежова сътрудник маршрутизиране и превключване, Сертифицирана мрежов сътрудник по защита на идентификационни данни**
- **Microsoft – Сертифициран разработчик на решения за уеб приложения, Сертифициран сътрудник решения за уиндоус сървър**
- **Професионални асоциации – Сертифициране за сътрудници за разработка на софтуер, Comptia's Security+, Comptia's A+ сертифициране, Comptia's Linux+**
- **С нестопанска цел – Сертифициран специалист по защита на информационните системи, Сертифициран мениджър по информационна сигурност, Сертифициране на професионално удостоверение за защита на софтуерен жизнен цикъл6**

3. Оценка

3.1 Оценка на знанията

Оценка подобна на викторина, основана на основното съдържание. Моля, маркирайте правилния отговор с удебелен шрифт, когато това се изисква. Включва 10 въпроса за вашия модул. Увеличава постепенно нивото на трудност.

Въпрос 1: Компютърният програмист е професионалната фигура, отговорна за писането на кодове, или *кодирането*, които инструктират компютъра, приложението или софтуерната програма за това как да работи.

[Вярно] [Невярно]

[6] Cote, J. (2021, August 19). What is Computer Programming? How to Become a Computer Programmer. SNHU.Edu. https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming

Въпрос 2: Процесът на кодиране може да накара компютъра само да работи. [Вярно] [Невярно]

Въпрос 3: Какви задачи трябва да владее компютърният програмист?

[Разрешаване на проблеми с компютърния софтуер] [Модифициране на софтуерни програми за подобряване на производителността] [Писане на код за компютърно програмиране] [Тестване на производителността на софтуера] [Всичко по-горе]

Въпрос 4: Частта от езика, която един компютър може да разбере, се нарича "бинарен". Как се нарича процесът на превод на програмен език в бинарен/двоичен?
[Кодиране] [Компилиране] [Отстраняване на грешки]

Въпрос 5: Какъв е правилният ред на задачите в процеса на програмиране?

[Кодиране на програмата, Тестване на програмата, Определяне на проблема, Планиране и проектиране на решението]
[Планиране и проектиране на решението, Определяне на проблема, Кодиране на програмата, Тестване на програмата]
[Определяне на проблема, Планиране и проектиране на решението, Кодиране на програмата, Тестване на програмата]

Въпрос 6: Какви са трите етапа на фазата на кодиране в рамките на процеса на програмиране?
[Тестване] [Кодиране] [Систематизиране] [Компилиране] [Отстраняване на грешки]

Въпрос 7: Какво напомнят програмните принципи на програмистите да направят при кодирането?
[да използват кодове, които други програмисти не могат да модифицират в бъдеще] [да използват сложни кодове, които засилват функциите] [да пишат лесни кодове] [да използвате чисти променливи, които остават силни след тестване]

Въпрос 8 (съчетайте): Съчетайте термините с дефинициите им.

Бинарен (двоичен): Базисна система от 2-числа, измислена от Готфилд Лайбниц, която е съставена само от две числа или цифри: 0 (нула) и 1 (едно).

ИКТ (ICT) Тази дума се отнася до технологии, които предоставят достъп до информация чрез телекомуникации.

Машинно обучение: клон на изкуствения интелект (ИИ) и компютърните науки, който се фокусира върху използването на данни и алгоритми, за да имитира начина, по който хората учат, постепенно подобрявайки точността си.

Аналогов: Прилагателно, което се отнася до механизъм или устройство, в което информацията е представена от непрекъснато променливи физически количества. Това е противоположното на дигитален.

Въпрос 9 (съчетайте): Съчетайте понятията с обясненията им.

Компютърен програмист: лице, отговорно да създаде инструкции за компютър, които да изпълни чрез писане и тестване на код, който дава възможност на приложения и софтуерни програми да работят успешно.

Компилиране: процес на превод на програмен език в бинарен.

Отстраняване на грешки: Процес, който включва идентифициране на всички грешки в програмата, благодарение на помощта на компилатора, преглед на оригиналната програма, идентифициране на грешките, коригиране на кода и компилиране отново.

Кодиране: процес на превод на дизайна във функционална програма, с използване на език за програмиране.

Тестване: процес, който включва преглед на програмата, за да се провери дали работи точно по начина, по който сте искали.

3.2 Оценка на уменията

Аналитично мислене е важна черта на един програмист. Нарича се също аналитично разсъждение, абстрактно мислене или критично мислене. Хората, които могат да мислят логично, са в състояние да анализират проблемите и да изработят решения. Това е не само ценно при разработването на програми, но е жизненоважно за всяка ситуация, която изисква рационална мисъл. Хора, които са логични:

- **анализира информация или ресурси, свързани със задача**

- внимателно наблюдавайки случващото се
- изучават информацията обективно, за да се определи дали тя е уместна или вярна
- съсредоточават се върху факти, не емоции
- разработва решения на проблеми въз основа на факти
- очертават ясно идеите, като ги разбиват на части
- обръщат внимание на подробностите
- тестват ефективността на дадено решение и правят преработки

Съществуват много упражнения и дейности, за практикуване на това умение. Някои от тях са много прости, като четене на книги, игра на мозъчни игри или присъединяване към клуб за дебати, за да поддържате мозъка си стимулиран. Ако сте програмист, лесен начин да подобрите уменията си за аналитично мислене е да следвате и имате предвид това ръководство за действие от 5 стъпки:

1. **Анализиране на проблема: включва събиране на информация, разглеждане на ресурси и определяне на пропуски в уменията или знанията;**
2. **Формулиране на план: събиране на идеи като при брейнсторминг и организиране на мислите си в план;**
3. **Разработване на код за решаване на проблема: започвате да пишете кода (познаването на компютърните езици е от съществено значение);**
4. **Оценка на решението и ревизиране на кода: Преглед на решението и определяне на области за подобрение, ако е необходимо;**
5. **Обосноваване на решения: представяне на доказателства, които обясняват защо програмата е приемливо решение.**

Още информация тук: https://www.technokids.com/blog/teaching-strategies/its-easy-to-improve-logical-thinking-with-programming/

4. Използвана литература

Cote, J. (2021, August 19). What is Computer Programming? How to Become a Computer Programmer. SNHU.Edu. https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming

Weinstein, J. (2021, January 24). What Is Coding? Coding Definition and Uses. Career Karma. https://careerkarma.com/blog/what-is-coding-used-for/
Writers, S. (2021, September 27). Guide to Programming Languages | ComputerScience.org. Get an Education the World Needs | ComputerScience.Org. https://www.computerscience.org/resources/computer-programming-languages/

Gallagher, J. (2021, May 4). The Most Popular Programming Languages. Career Karma. https://careerkarma.com/blog/top-programming-languages-2021/

Techopedia. (2014, July 30). Computer Programmer. Techopedia.Com.

https://www.techopedia.com/definition/6589/computer-programmer

The Programming Process. (2020). Cs.Bham.Ac.Uk.

https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html