

TVORBA, TESTOVÁNÍ A PROVOZ SOFTWARE

VYSVĚTLENÍ A PŘÍKLADY K NOVÉMU RVP PRO SOV

Výstup RVP

Žák určí, zda je daný postup algoritmem; vysvětlí daný algoritmus, program.

Učivo (RVP)

pojem algoritmus; vlastnosti algoritmu; různé zápisy algoritmů

Vysvětlení

Žáci se učí identifikovat klíčové vlastnosti algoritmů, rozlišovat různé způsoby zápisu algoritmů a efektivně komunikovat principy fungování algoritmu. Tímto se rozvíjí jejich schopnost určit, zda je daný postup algoritmem, a také dovednost vysvětlit fungování konkrétního algoritmu či programu.

Rozklad výsledku vzdělávání

Určení, zda je daný postup algoritmem

Žák chápe algoritmus jako sadu instrukcí nebo pravidel pro řešení problému nebo provedení úkolu. Seznámí se s vlastnostmi algoritmu, jako jsou jasnost, jednoznačnost, konečnost, efektivita a schopnost vyprodukovat výstup po konečném počtu kroků. Pracuje s různými příklady postupů a rozlišuje, zda se jedná, nebo nejedná o algoritmy.

Analýza a pochopení algoritmu

Žák analyzuje konkrétní algoritmus. Je schopen identifikovat jednotlivé kroky algoritmu a popsat, co každý krok dělá. Zná praktické příklady algoritmů, rozdělí tyto algoritmy na jednotlivé kroky a porozumí jejich účelu.

Různé zápisy algoritmů

Žák rozumí tomu, že algoritmy lze zapisovat různými způsoby. Rozliší pseudokód, diagramy toku a základní struktury programovacích jazyků. S využitím konkrétních ukázek se seznámí s tím, jak lze jeden algoritmus přepsat z jedné formy do druhé.

Vysvětlení daného algoritmu, programu

Žák je schopen vysvětlit daný algoritmus nebo program. S využitím praktických cvičení porozumí problematice, představí analyzovaný algoritmus a objasní jeho funkci a účel.

PRAKTICKÝ PŘÍKLAD 1

GASTRONOMIE

Určení, zda je daný postup algoritmem

Pečení chleba. V prvním kroku představit proces pečení chleba jako sérii kroků, které musí být dodrženy, aby byl výsledek úspěšný – například měření ingrediencí, míchání, kynutí těsta a pečení. Porozumět tomu, že tento postup je algoritmem, protože obsahuje jasné, jednoznačné a konečné kroky vedoucí k předvídatelnému výstupu, což je upečený chléb.

Analýza a pochopení algoritmu

Následně podrobně procházet každý krok procesu pečení chleba. Diskutovat o tom, proč je důležité dodržet určité množství surovin, proč těsto musí dostatečně kynout a jaké teploty a časy jsou pro pečení klíčové. Vést k porozumění toho, že každý krok je pro dosažení dobrého výsledku důležitý, pochopit logiku jednotlivých částí procesu.

Různé zápisy algoritmů

Seznámit se s různými způsoby záznamu procesu pečení chleba a možnostmi jeho předávání – například prostřednictvím tradičního receptu, grafického diagramu procesu či videonávodu. Identifikovat výhody jednotlivých metod a určovat, který ze způsobů je pro danou situaci/potřebu nejjasnější a nejpřístupnější.

Vysvětlení daného algoritmu, programu

V posledním kroku je úkolem představit celý proces pečení chleba spolužákům. Popsat jednotlivé kroky, vysvětlit, proč jsou důležité a co se stane, pokud nebudou správně provedeny. Prostřednictvím tohoto praktického nácviku upevnit pochopení algoritmu a rozvíjet schopnost předávat informace jasně a strukturovaně.

Určení, zda je daný postup algoritmem

Proces výroby glazovaného výrobku. V prvním kroku je třeba objasnit, že proces výroby glazované keramiky může být chápán jako algoritmus. Tento proces zahrnuje specifické kroky: příprava hlíny, tvarování výrobku, první vypalování, aplikace glazury a druhé vypalování. Každý z těchto kroků je jasně definován a je nezbytný pro dosažení konečného produktu, což je glazovaný keramický výrobek.

Analýza a pochopení algoritmu

Poté detailně prozkoumat každý krok procesu. Naučit se, jak správně připravit hlínu, jaké techniky tvarování použít pro vytvoření žádaného tvaru a jaké jsou klíčové aspekty při vypalování. Seznámit se s různými typy glazur a metodami jejich aplikace. Důraz je třeba klást na pochopení, jak každý krok ovlivňuje kvalitu a vzhled finálního produktu.

Různé zápisy algoritmů

Seznámit se s různými způsoby dokumentace procesu výroby keramiky. Ty mohou zahrnovat písemné instrukce, sériové fotografie každého kroku nebo dokonce videotutoriály. Porozumět tomu, že tyto záznamy pomáhají zachytit detaily procesu a slouží jako reference pro budoucí projekty nebo pro nové žáky v oboru.

Vysvětlení daného algoritmu, programu

Na závěr je zadáno celý proces výroby keramiky prezentovat. Popsat jednotlivé kroky, jejich důležitost a vliv na konečný výrobek. Tímto jsou upevňovány nejen znalosti, ale také rozvíjena schopnost komunikovat komplexní postupy jasně a efektivně.

Určení, zda je daný postup algoritmem

Proces barvení vlasů. Nejprve je třeba porozumět tomu, že proces barvení vlasů může být považován za algoritmus, protože zahrnuje specifické, přesně definované kroky. Tyto kroky zahrnují konzultaci s klientem, výběr barvy, přípravu vlasů, aplikaci barvy, čekání na působení, oplachování barvy a styling. Každý krok je nezbytný pro dosažení žádaného výsledku – rovnoměrně obarvených vlasů.

Analýza a pochopení algoritmu

Následuje podrobná analýza každého kroku procesu barvení. Je potřeba zjistit, jak důležitá je přesná konzultace pro pochopení přání klienta a jaké faktory ovlivňují výběr barvy, například typ a stav vlasů klienta. Prozkoumat, jak správně míchat barvivo s aktivátorem a jaké techniky aplikace zajišťují rovnoměrné pokrytí bez poškození vlasů. Porozumět návaznosti jednotlivých kroků.

Různé zápisy algoritmů

Seznámit se s různými způsoby, kterými může být proces barvení dokumentován. To může zahrnovat checklisty pro každý krok, instruktážní videa ukazující techniky aplikace nebo fotografické dokumentace před barvením a po něm. Taková dokumentace pomáhá zajistit konzistenci a kvalitu služeb, také může být podkladem pro zákazníka při další návštěvě pro srovnání a pro evidenci již aplikovaných odstínů, tzv. tvorba zákaznickova portfolia.

Vysvětlení daného algoritmu, programu

V posledním kroku celý proces barvení vlasů prezentovat. Vysvětlit kroky, jejich důležitost a dopady na konečný výsledek. Prezentace je důležitá pro upevnění znalostí a rozvoj efektivní komunikace a schopnosti aplikovat naučené dovednosti.



TVORBA, TESTOVÁNÍ A PROVOZ SOFTWARE

VYSVĚTLENÍ A PŘÍKLADY K NOVÉMU RVP PRO SOV

Výstup RVP

Žák rozdělí problém na menší části, rozhodne, které je vhodné řešit algoritmicky, své rozhodnutí zdůvodní; sestaví a zapíše algoritmy pro řešení problému.

Učivo (RVP)

rozdělení problému na části; identifikace návazností dat, opakujících se vzorů a míst pro rozhodování

Vysvětlení

Rozvíjené **dovednosti**, tedy **rozdělit složité problémy na menší**, spravovatelné části, **identifikovat kroky vhodné pro algoritmické řešení a efektivně sestavit a zapsat algoritmy**, posilují nejen **technické schopnosti**, ale i **schopnost jasně a logicky formulovat kroky potřebné k vyřešení problému**. **Důraz je třeba klást na praktické aplikace v různých oborech**, což žákům umožní lépe pochopit a implementovat teoretické koncepty do skutečných pracovních procesů.



Rozklad výsledku vzdělávání

Identifikace problému a jeho rozdělení na menší části

Žák rozloží složitý problém na menší, spravovatelné části a chápe, že tím může získat lepší přehled a strukturovat přístup k řešení. Seznámí se s praktickými ukázkami a příklady toho, jak rozdělit komplexní softwarový projekt na jednotlivé moduly nebo funkcionality. Rozumí tomu, že tento postup pomáhá lépe organizovat práci a rozdělit zodpovědnosti mezi tým.

Rozhodnutí o algoritmickém řešení vybraných částí

Žák rozlišuje, které části problému jsou vhodné pro algoritmické řešení. Identifikuje opakující se vzory a procesy, které lze efektivně automatizovat nebo standardizovat. Chápe, že toto rozhodování je zásadní pro optimalizaci zdrojů a zvyšování efektivity vývoje softwaru.

Sestavení a zápis algoritmů

Žák umí sestavit a správně zapsat algoritmus. Získá praktickou zkušenost se psaním pseudokódu a skutečného kódu v různých programovacích jazycích. Rozumí tomu, jak se algoritmické řešení převádí do funkčních programových bloků.

PRAKTICKÝ PŘÍKLAD 1

NÁBYTKÁŘSTVÍ

Identifikace problému a jeho rozdělení na menší části

Výroba dřevěných stolů (sestavení a programování CNC stroje). Nejprve je třeba rozdělit proces výroby stolu na základní operace. Rozpoznat, že celý proces začíná návrhem stolu, výběrem materiálu a nastavením CNC stroje. Každá z těchto částí vyžaduje specifické kroky, jako je výběr typu dřeva, rozměrů, designu a parametrů řezání.

Rozhodnutí o algoritmickém řešení vybraných částí

Vybrat vhodné řešení problému a v následujícím kroku jej algoritmi- zovat.

Sestavení a zápis algoritmů

Sestavit algoritmus pro řídicí soft- ware CNC stroje. Zapsat jednotlivé kroky programu, který bude řídit stroj tak, aby automaticky vykonal potřebné úkony podle designu stolu. Do algoritmu zahrnout instrukce pro pohyb nástroje na základě designo- vých souborů, které byly předem vytvořeny v CAD softwaru.

PRAKTICKÝ PŘÍKLAD 2

CUKRÁŘSTVÍ

Identifikace problému a jeho rozdělení na menší části

Potahovaný dort. Při tvorbě pota- hovaného dortu je nejprve potřeba rozdělit celý proces na základní kroky: příprava těsta, pečení korpu- su, příprava krému, sestavení dortu a potažení dortu fondánem. Každý z těchto kroků vyžaduje specifické suroviny a techniky, například správné míchání ingrediencí pro těsto, kontrolu teploty pečení nebo techniku roztírání krému a aplikace fondánu.

Rozhodnutí o algoritmickém řešení vybraných částí

Dále je nutné určit, které části pro- cesu mohou být standardizovány nebo zjednodušeny. I když v cukrář- ství není běžné použití tradičního programování, je možné algoritmic- ky přistupovat k recepturám a po- stupům. Například přesné měření surovin standardizovat pomocí vah s digitálním displejem, což zajistí konzistenci výsledků. Stejně tak lze standardizovat proces ochlazová- ní a krájení korpusu pro dosažení ideální teploty a velikosti vrstev.

Sestavení a zápis algoritmů

Následně sestavit „recept-algorit- mus“, který podrobně popíše kroky výroby od přípravy až po dokončení dortu. Algoritmus může zahrnovat časový harmonogram pro každou část procesu, což pomáhá zajistit provedení všech kroků ve správném pořadí a čase. Tento postup lze také zapsat jako kontrolní seznam a použít k ověření, že žádný krok není vynechán a výsledný produkt je vizuálně i chuťově atraktivní.

PRAKTICKÝ PŘÍKLAD 3

ZAHRADNICTVÍ

Identifikace problému a jeho rozdělení na menší části

Plánování a realizace zahradního designu. Proces zahájit analýzou prostoru, hodnocením typu půdy, světelných podmínek a stávající vegetace. V dalším kroku určit funkční zóny zahrady, jako jsou od- počinkové zóny, zóny pro pěstování rostlin nebo hřiště. Každá z těchto částí vyžaduje specifické plánování a zohlednění individuálních potřeb.

Rozhodnutí o algoritmickém řešení vybraných částí

Rozhodnout o optimálním rozmís- tění rostlin a prvků zahradní archi- tektury. Využít software pro návrh krajiny, který umožňuje vizualizovat a plánovat rozmístění prvků zahrad- ního designu. Tento software může být programován tak, aby respekto- val zásady dobrého designu, jako jsou symetrie, repetice nebo kon- trast, a zároveň zohlednil optimální podmínky pro růst rostlin, jako jsou světlo, vlhkost a typ půdy. Zvážit návrh postupu výsadby navržené architektury.

Sestavení a zápis algoritmů

Vytvořit detailní plán implementace návrhu umístění architektonických prvků a rostlin podle druhů. Plán lze popsat jako krok za krokem algorit- mus pro samotnou realizaci zahrady, zahrnující přípravu terénu, výsadbu, instalaci závlahového systému a další prvky zahradního designu. Algoritmus může obsahovat i časo- vou osu pro realizaci jednotlivých kroků, což zahrnuje specifické úkoly, jako je příprava půdy, výsadba rostlin v nevhodnější sezóně a sledování růstu rostlin.

TVORBA, TESTOVÁNÍ A PROVOZ SOFTWARE

VYSVĚTLENÍ A PŘÍKLADY K NOVÉMU RVP PRO SOV

Výstup RVP

Žák zobecní řešení pro širší třídu problémů; ověří správnost, najde a opraví případnou chybu v algoritmu.

Učivo (RVP)

zadání úlohy, vstup, výstup, podmínky řešení; volba nástroje podle zadání úlohy

Vysvětlení

Schopnost analyzovat zadání úlohy, vybírat vhodné nástroje a navrhnout efektivní algoritmy přispívá k rozvoji kritického myšlení a schopnosti reflexe (sebereflexe). **Žáci ověřují správnost algoritmu pomocí testování, identifikují a opraví případné chyby.** Tento proces zahrnuje rovněž důkladnou dokumentaci a evaluaci algoritmu.

Rozklad výsledku vzdělávání

Analýza zadání a volba nástrojů

Žák identifikuje v zadání úlohy klíčové vstupy, očekávané výstupy a podmínky řešení. Vybere nejvhodnější programovací nástroj nebo software, který odpovídá specifikacím problému. Při volbě správného nástroje posoudí jeho funkcionality, efektivitu a kompatibilitu s daným úkolem.

Navrhování a generalizace algoritmu

Žák umí zobecnit navržený algoritmus tak, aby byl aplikovatelný na širší třídu podobných problémů. Chápe, že tento postup mu umožňuje flexibilně přizpůsobit řešení novým výzvám.

Testování a oprava chyb

Žák provádí testy s různými vstupy s cílem ověřit, že výstupy algoritmu jsou správné. Při zjištění nesrovnalostí nebo chyb analyzuje chování algoritmu, lokalizuje problémové části a zajistí potřebné opravy. Rozumí významu důkladného testování jako klíčového kroku pro zajištění správné funkčnosti a spolehlivosti softwaru.

Dokumentace a evaluace

Žák zaznamená postupy, použité nástroje a způsoby řešení s cílem usnadnit další údržbu a rozvoj softwaru. Vyhodnotí efektivitu algoritmu a porovná jej s alternativními řešeními.

PRAKTICKÝ PŘÍKLAD 1

OBRÁBĚČ KOVŮ

Analýza zadání a volba nástrojů

Soustružení kovové hřídele. Nejprve je třeba připravit technický postup podle výkresové dokumentace a upřesnit nástroje a materiál. Dále pak zvážit použití softwarových nástrojů.

Navrhování a generalizace algoritmu

V dalším kroku vytvořit algoritmus pro zadanou třídu úloh.

Testování a oprava chyb

Podle možností vytvořit program v testovacím prostředí. Provést test a opravit chyby či kolize nebo vyrobit kus modelu 3D a zjistit případné chyby.

Dokumentace a evaluace

Zhotovit dokumentaci postupu, případně výkres, a společně zhodnotit práci.

Analýza zadání a volba nástrojů

Vytvoření návrhu 3D modelu pece nebo kamen. V zadání specifikovat možnosti tvaru kamen a materiály. Na základě toho vybrat jednoduchý CAD software, který je dostupný a vhodný pro tyto účely.

Navrhování a generalizace algoritmu

Nastavit pro CAD software jednotlivé segmenty ramen tak, aby odpovídaly specifickým požadavkům kamnářského designu. To zahrnuje přidání šablon pro běžné tvary a materiály, což ulehčí návrhářský proces a zefektivní práci.

Testování a oprava chyb

Otestovat na modelu předpokládanou funkcionalitu. Případné chyby v modelování opravit tak, aby se zajistilo, že výsledný model přesně odpovídá požadavkům a je technicky proveditelný.

Dokumentace a evaluace

Po úpravách vytvořit základní popis modelu kamen pro případného zákazníka. Provést finální hodnocení s mistrem. Diskutovat, jak software zlepšil návrhový proces a zda jsou potřebné další úpravy.

Analýza zadání a volba nástrojů

Vizualizace a plánování dekorací. Zakázkou je kompletní dekorace svatebního sálu, v rámci které klient požaduje reflektovat specifické barvy a téma ve všech aspektech od květinových aranžmá po stolní výzdobu. Nejprve je třeba vybrat pokročilý grafický software umožňující 3D vizualizaci, který disponuje knihovnou reálných textur a materiálů. Cílem toho je usnadnit klientovi pochopení a schválení finálního návrhu před realizací.

Navrhování a generalizace algoritmu

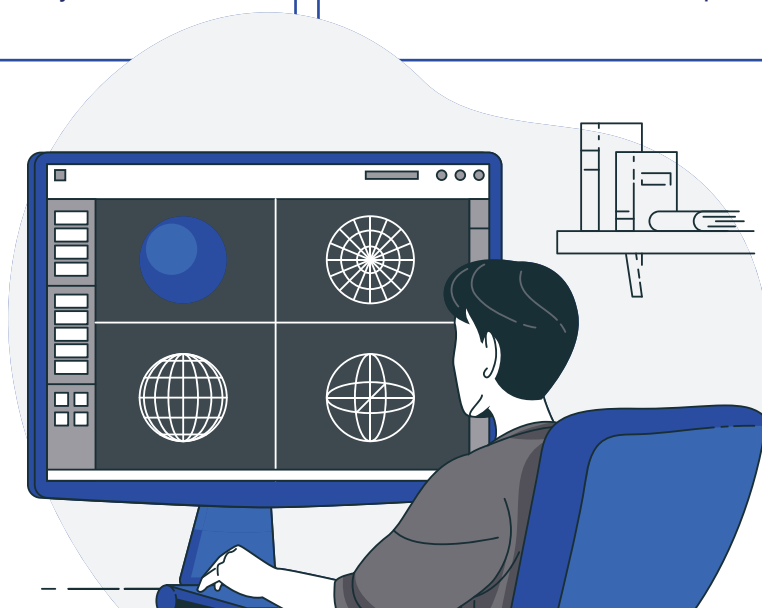
Sestavit v grafickém softwaru model svatebního sálu. Začít vložením a úpravou jednotlivých dekorativních prvků podle požadavků klienta. Využít funkcionalit softwaru (možnost snadno měnit barvy a materiály v reálném čase) a tím si usnadnit experimentování s různými kombinacemi a okamžité přizpůsobení návrhu.

Testování a oprava chyb

Po dokončení prvotního návrhu představit 3D vizualizaci klientovi. Společně projít návrh a diskutovat o možných změnách nebo vylepšeních. Na základě zpětné vazby upravit návrh tak, aby odpovídal přesným požadavkům klienta. Opětovně testovat vizualizaci s cílem zajistit bezchybnost a vizuální jednotnost všech aspektů dekorace.

Dokumentace a evaluace

Finální verzi návrhu zdokumentovat v podobě detailních nákreseů a seznamu materiálů potřebných pro realizaci. Zaznamenat změny provedené na základě klientových přání. Po realizaci dekorace spolu s klientem vyhodnotit efektivitu softwaru ve schopnosti předpovědět výsledný vizuální efekt a v realizaci všech dekorativních prvků podle očekávání.



TVORBA, TESTOVÁNÍ A PROVOZ SOFTWARE

VYSVĚTLENÍ A PŘÍKLADY K NOVÉMU RVP PRO SOV

Výstup RVP

Žák hodnotí algoritmy podle různých hledisek, porovná a vybere pro řešený problém ten nejvhodnější; vylepší algoritmus podle zvoleného hlediska.

Učivo (RVP)

drůhy chyb; chybové hlášky; hlášení a evidence závad; způsoby testování programu

Vysvětlení

Komplexní **dovednosti v oblasti programování a údržby softwaru podporují schopnost testovat algoritmus a identifikovat a opravit jeho případné chyby.** To přirozeně vyžaduje, že jsou žáci nejen schopni identifikovat a porozumět specifikacím problému, ale také vyhodnotit různé algoritmy na základě klíčových kritérií, jako jsou efektivita a složitost. **Žáci se proto musí naučit tyto algoritmy porovnat, vybrat ten nejvhodnější pro daný problém a ten následně optimalizovat podle zvolených hledisek.**

Rozklad výsledku vzdělávání

Pochopení a definice problému

Žák analyzuje problém a identifikuje klíčové aspekty a požadavky, které musí řešení splňovat. Uvědomuje si, že pochopení problému je základem pro všechny následující kroky v procesu vývoje algoritmu.

Hodnocení a porovnání algoritmů

Žák zná způsoby, jak posuzovat různé algoritmy podle kritérií jako efektivita, složitost a spotřeba zdrojů. Má praktickou zkušenost s aplikací různých algoritmů na definovaný problém a s porovnáváním jejich výkonnosti a vhodnosti.

Výběr a optimalizace algoritmu

Na základě porovnání různých algoritmů je žák schopen vybrat ten, který nejlépe odpovídá požadavkům daného problému. Věnuje pozornost jeho vylepšení a optimalizaci s cílem zvýšit efektivitu algoritmu a snížit nároky na systémové zdroje. Při optimalizaci zvažuje úpravy pro zvýšení rychlosti, redukci paměťové náročnosti nebo zlepšení čitelnosti a udržitelnosti kódu.

Způsoby testování algoritmu a řešení závad

Žák vybírá vhodný způsob testování, provede jej, ověří jeho funkčnost a identifikuje případné chyby. Seznámí se s metodami pro hlášení a evidenci závad, včetně rozpoznávání běžných typů chyb a chybových hlášek. Dokáže efektivně detekovat problémy a zavádět nápravy.

PRAKTICKÝ PŘÍKLAD 1

MALOOBCHOD

Pochopení a definice problému

Optimalizace algoritmu pro správu skladových zásob. Zadáním je vyřešit snížení množství přebytečných zásob bez rizika, že dojde k jejich nedostatku. Cílem je tedy určit algoritmus, který nejlépe vyhovuje potřebám obchodu s ohledem na sezónní výkyvy a měnící se poptávku zákazníků.

Výběr a optimalizace algoritmu

Na základě výsledků porovnání vybrat algoritmus, který se jeví jako nejvhodnější – například EOQ pro schopnost efektivně vypočítat optimální množství objednávky na základě nákladů a poptávky. Algoritmus nebo jeho parametry upravit tak, aby reflektoval specifika obchodu, jako jsou speciální akce, sezónní prodeje a možná dodavatelská zpoždění.

Hodnocení a porovnání algoritmů

Seznámit se s několika algoritmy pro správu zásob, jako jsou Just-in-Time (JIT), Economic Order Quantity (EOQ) a First-In-First-Out (FIFO). Vyhodnotit každý algoritmus podle jeho schopností minimalizovat náklady na skladování a zároveň zajistit dostatečné množství zboží na skladě. V praxi to znamená porovnat historická data o prodejích a objednávkách a tím zjistit, který systém nejlépe odpovídá dynamice obchodu.

Způsoby testování algoritmu a řešení závad

Nakonec implementovat upravený algoritmus a sledovat jeho výkonnost v reálném provozu. Během testovacího období pravidelně kontrolovat, zda dochází k nedostatkům nebo přebytkům zásob, a identifikovat případné chyby v algoritmu. Současně zaznamenávat chybové hlášky systému a jejich řešení. Díky tomuto kroku se zvyšuje přesnost a efektivita správy zásob, čímž se snižují celkové náklady obchodu.

Pochopení a definice problému

Optimalizace procesu měření a řezání materiálu pro montáž vodních žlabů a okapů. Zadáním je efektivně změřit a připravit materiál tak, aby byl využitelný na různé typy střech a fasád. To zahrnuje správné dimenzování a minimalizaci odpadu. Porozumět problému, který spočívá v optimalizaci procesu měření a řezání kovových listů na míru.

Hodnocení a porovnání algoritmů

Seznámit se s různými metodami měření a značení materiálu, včetně manuálních technik a digitálních nástrojů pro přesné měření. Metody posoudit a ohodnotit podle přesnosti, rychlosti a snadnosti použití a efektivity.

Výběr a optimalizace algoritmu

Na základě výsledků hodnocení vybrat metodu, která nejlépe vyhovuje konkrétním požadavkům projektu. V případě potřeby metodu dále upravit pro zvýšení efektivity, například přizpůsobením měřicích nástrojů pro specifické typy materiálů a střešních konstrukcí.

Způsoby testování algoritmu a řešení závad

Po implementaci vybrané metody provést několik testovacích řezů a zkontrolovat, zda výsledky odpovídají požadovaným specifikacím. Jakékoliv nesrovnalosti zaznamenat a vyhodnotit s cílem dále vylepšit proces a snížit množství odpadu materiálu.

Pochopení a definice problému

Výroba večerních šatů. Úkolem je navrhnout a ušít večerní šaty, které dokonale sedí zákazníci a zároveň splňují její estetické představy. Klíčovým problémem je efektivně spojit požadavky na design s technickými aspekty šití, aby bylo dosaženo optimálního výsledku s minimálním odpadem materiálu.

Hodnocení a porovnání algoritmů

Prozkoumat různé techniky měření a stříhu látek, od tradičních ručních metod po modernější digitálně podporované postupy. Tyto metody hodnotit podle přesnosti, efektivity a schopnosti adaptace na složité designy.

Výběr a optimalizace algoritmu

Na základě výsledků hodnocení vybrat metodu, která nejlépe vyhovuje specifickým potřebám zakázky. Vybranou techniku podle potřeby upravit pro zlepšení práce s konkrétními typy látek a složitějšími stříhy. Porozumět tomu, že takový postup zefektivní proces a zvyšuje kvalitu výsledného produktu.

Způsoby testování algoritmu a řešení závad

Po implementaci nového postupu následuje série zkoušek s cílem zkontrolovat, jak dobře šaty sedí a jak přesně odpovídají požadovaným specifikacím. Problémy, jako jsou nepřesnosti ve stříhu nebo nedostatečná funkčnost švu, identifikovat a opravit. Tím zaručit vysokou úroveň kvality hotového oděvu.



TVORBA, TESTOVÁNÍ A PROVOZ SOFTWARE

VYSVĚTLENÍ A PŘÍKLADY K NOVÉMU RVP PRO SOV

Výstup RVP

Žák sestaví přehledný program v blokově orientovaném nebo textovém jazyce, program otestuje a optimalizuje.

Učivo (RVP)

zápis algoritmu vhodnou formou (například blokové schéma, přirozené a formální jazyky, skriptovací a programovací jazyk); základní koncepce tvorby programů (například proměnná a datový typ, řídicí příkazy, cykly); návrh programu

Vysvětlení

Proces zahrnuje formulaci algoritmu, seznámení se základy programovacího jazyka a sestavení i následné testování a optimalizaci programu. Tyto kroky připravují žáky na reálné výzvy IT oblasti, zlepšují jejich schopnosti analyzovat, programovat a efektivně řešit technologické problémy.

Rozklad výsledku vzdělávání

Zápis algoritmu vhodnou formou

Žák si osvojí různé formy zápisu algoritmů, jako jsou bloková schémata a formální jazyky. Rozumí tomu, že výběr vhodného typu zápisu pomáhá lépe pochopit a vizualizovat problémy před samotným programováním.

Základy programovacího jazyka

Žák má znalosti o syntaxi a základních konceptech programovacího jazyka, jako jsou proměnné, datové typy, řídicí příkazy a cykly.

Sestavení programu

Žák prakticky aplikuje naučené koncepty při tvorbě programu, který musí splnit specifikované požadavky.

Testování a optimalizace programu

Žák je schopen efektivně testovat software a identifikovat oblasti pro zlepšení, jako jsou rychlost a spotřeba paměti. Rozumí tomu, že optimalizace zvyšuje kvalitu a efektivitu programu.

PRAKTICKÝ PŘÍKLAD 1

AUTOMOBILOVÝ PRŮMYSL

Zápis algoritmu vhodnou formou

Zpracování dat získaných z měřicího softwaru. Použít blokové schéma k vizualizaci postupu, jak software skenuje různé komponenty vozidla a identifikuje potenciální problémy.

Základy programovacího jazyka

Následně si osvojit základy textového programovacího jazyka, jako je Python, který je vhodný pro začátečníky a efektivní pro tento typ aplikací. Seznámit se s proměnnými, datovými typy a řídicími příkazy a ty použít k napsání základního programu pro diagnostiku. Vytvořit program tak, aby byl schopen načítat data z automobilových senzorů a interpretovat je.

Sestavení programu

Po osvojení teoretických znalostí sestavit reálný diagnostický program. Program musí správně komunikovat s obsluhou vstupních dat. Dle předem definovaných parametrů určovat, zda jsou hodnoty v normě, nebo zda signalizují potenciální problém.

Testování a optimalizace programu

Na závěr program testovat na skutečných datech vozidel. Ověřit, zda program správně detekuje a diagnostikuje problémy. Následně analyzovat výkon programu a hledat možnosti, jak zlepšit jeho rychlost a spolehlivost, například optimalizací kódu nebo přizpůsobením pro různé modely vozidel.

Zápis algoritmu vhodnou formou

Plánování a sledování regeneračních procedur sportovce. Nejprve je třeba použít blokové schéma pro návrh procesu, jak software eviduje jednotlivá sezení, monitoruje pokrok sportovce a identifikuje potřebné úpravy v terapii. Tento přehledný zápis umožňuje lépe porozumět funkcionalitě programu před jeho vývojem.

Základy programovacího jazyka

Poté se seznámit s jednoduchými principy programovacího jazyka, jako je JavaScript, který může být využit pro tvorbu jednoduchých webových aplikací. Osvojit si základní programovací koncepty, jako jsou proměnné, datové typy a jednoduché řídicí příkazy. Tyto znalosti využít k vytvoření základního rozhraní, které umožní zaznamenávat informace o klientech a jejich sezeních.

Sestavení programu

Po zvládnutí teoretických základů sestavit jednoduchý webový program, který mu umožní spravovat seznam klientů, jejich plánovaná a absolvovaná sezení a případné poznámky k jejich zdravotnímu stavu. Program vytvořit jako interaktivní s možností přidávat, měnit či odebírat data podle aktuálních potřeb.

Testování a optimalizace programu

V závěrečné fázi testovat funkčnost programu ve skutečném provozu s cílem zjistit, jak dobře software slouží pro evidenci a plánování sezení. Na základě zpětné vazby od klientů a vlastních zkušeností s programem hledat možnosti, jak zlepšit uživatelské rozhraní a zefektivnit práci s daty.

Zápis algoritmu vhodnou formou

Vytvoření algoritmu pro systém automatického zavlažování. Využít blokové schéma pro návrh toho, jak systém detekuje úroveň vlhkosti v půdě a podle toho aktivuje zavlažování.

Základy programovacího jazyka

Seznámit se s blokově orientovaným programovacím jazykem, jako je Scratch nebo Blockly, který umožňuje sestavovat programy táhnutím a umísťováním vizuálních bloků. Naučit se základní programovací struktury, jako jsou podmínky a smyčky, které použije pro ovládní senzorů vlhkosti a zavlažovacího systému.

Sestavení programu

Sestavit vizuální program, který řídí zavlažování zahrady na základě dat z vlhkostních senzorů. Porozumět tomu, že takový program zautomatizuje proces zavlažování a zefektivní použití vody.

Testování a optimalizace programu

V poslední fázi testovat efektivitu programu ve skutečném zahradním prostředí, kontrolovat správnost funkcí a shromáždit zpětnou vazbu k účinnosti zavlažování. Dále provést úpravy a optimalizace k dosažení nejlepší možné efektivity.



TVORBA, TESTOVÁNÍ A PROVOZ SOFTWARE

VYSVĚTLENÍ A PŘÍKLADY K NOVÉMU RVP PRO SOV

Výstup RVP

Žák používá základní programové konstrukce.

Učivo (RVP)

základní koncepce tvorby programů (například proměnná a datový typ, řídicí příkazy, cykly); nápověda a licence programu

Vysvětlení

Žáci si osvojí základní koncepty a techniky programování, jako jsou proměnné, datové typy, řídicí příkazy a cykly. Naučí se, jak tyto programové konstrukce efektivně používat pro vytváření, testování a provozování softwaru. Porozumí významu nápovědy a licenčních podmínek programů. Cílem je vybavit žáky dostatečnými znalostmi a dovednostmi pro samostatné řešení základních programovacích úloh a porozumění právním aspektům používání softwaru.

Rozklad výsledku vzdělávání

Porozumění pojmu proměnná a datový typ

Žák chápe proměnné jako základní stavební kameny pro uchování a manipulaci s daty v programu. Rozumí různým datovým typům, jako jsou celá čísla, desetinná čísla nebo text.

Práce s řídicími příkazy

Žák umí používat základní řídicí příkazy, jako jsou *if*, *else* a *switch*. Je si vědom toho, že tyto příkazy umožňují programu reagovat na různé vstupy nebo stavy proměnných.

Implementace cyklů

Žák ovládá cykly, jako jsou *for*, *while* a *do-while*. Rozumí, že jsou zásadní pro opakování určitých částí kódu. Uvědomuje si, že ovládání cyklů je klíčové pro efektivní zpracování dat, iteraci a automatizaci procesů.

Využití nápovědy a porozumění licencím softwaru

Žák zná a dokáže efektivně využívat nápovědu k programovacím jazykům. Rozumí základním principům licencování softwaru.

PRAKTICKÝ PŘÍKLAD 1

MALÝ VÝROBNÍ PODNIK

Porozumění pojmu proměnná a datový typ

Evidence skladových zásob. Nejprve je třeba nastavit základní strukturu databáze skladu, kde každá položka má své unikátní ID (celé číslo), název (text), množství na skladě (celé číslo) a jednotkovou cenu (desetinné číslo). Tvorba struktury pomáhá pochopit, jak různé datové typy slouží k efektivní organizaci informací v softwarové aplikaci.

Práce s řídicími příkazy

Následně vytvořit funkce v aplikaci, které umožní přidávání a odebrání zásob z databáze. Použití řídicích příkazů jako *if* a *else* zajistí, že množství na skladě nemůže klesnout pod určitou hodnotu a že jsou zadány všechny potřebné informace před přidáním položky do databáze.

Implementace cyklů

Poté pro zobrazení celkové hodnoty skladových zásob vytvořit smyčku, která projde všechny položky v databázi a vypočítá celkovou hodnotu zásob tím, že vynásobí množství každé položky její jednotkovou cenou. Tento proces demonstruje, jak cykly umožňují efektivně zpracovávat opakované úlohy v rámci softwaru.

Využití nápovědy a porozumění licencím softwaru

Na závěr použít nápovědu pro užití softwaru, dokumentaci k programovacímu jazyku a zvolit typ licencování daného softwaru, například podle počtu uživatelů nebo podle délky použití programu.

Porozumění pojmu proměnná a datový typ

Optimalizace procesu pečení a správy zásob. V prvním kroku vytvořit digitální evidenční systém pro suroviny potřebné k pečení. Pro každou surovinu specifikovat ID, název, množství a datum expirace. Tento systém umožní pekařům snadno identifikovat a spravovat suroviny v pekárně.

Práce s řídicími příkazy

Následně do systému implementovat funkce pro aktualizaci stavu surovin při jejich použití nebo doplnění. Pomocí řídicích příkazů *if* a *else* zajistit, že systém upozorní na nedostatek surovin nebo blížící se expiraci, což umožní pekaři předcházet plýtvání a zajistit efektivitu.

Implementace cyklů

Pro automatické zpracování objednávek a plánování produkce použít smyčku, která projde všechny aktuální objednávky a vypočítá potřebné množství surovin na základě receptur. Tím se minimalizují chyby v množstvích a zvyšuje se produktivita pekárny.

Využití nápovědy a porozumění licencím softwaru

Použít nápovědu k programu a zohlednit omezení daná licencí.

Porozumění pojmu proměnná a datový typ

Monitorování a plánování denní péče o pacienty. Nejprve je třeba vytvořit systém pro evidenci každodenních činností pacientů, jako je ranní hygiena, podávání medikace, převazy a evidence stavu vylučování a stavu výživy pacientů. Každé činnosti přiřadit unikátní ID, popis, časový údaj, stav dokončení a počet opakování. Tento systém umožní ošetřovatelům efektivně sledovat zdravotní stav a plánovat denní péči o pacienty, předcházet chybám v podávání léků ve správný čas a ve správném množství. Systém eviduje množství skladových zásob jednotlivých léků.

Práce s řídicími příkazy

Rozšířit systém o funkcionalitu pro aktualizaci stavu dokončení většiny činností. Použitím řídicích příkazů *if* a *else* zajistit, že žádná z plánovaných aktivit není přehlédnuta a vše je zaznamenáno v reálném čase. To pomáhá zajišťovat kvalitní péči a dodržovat stanovený režim pacienta.

Implementace cyklů

Pomocí cyklů automaticky generovat denní a týdenní přehledy činností, které pomáhají ošetřovatelům při předávání směn a koordinaci s ostatním personálem. Nechat smyčkou projít všechny činnosti a zjistit, které byly dokončeny a které vyžadují další pozornost.

Využití nápovědy a porozumění licencím softwaru

Reflektovat licenční podmínky softwaru a dodržování pravidel ochrany osobních a citlivých údajů pacientů. Efektivně využívat nápovědu k programům a zohledňovat omezení daná licencí.

