## Dokumentation Mathe Projekt (V01.1)

|  |
| --- |
| Sortieren eines eindimansionalen Arrays  Benutzung:  Die Funktion wird folgendermaßen aufgerufen:  ArraySort(<arrayLength>, &<array[0]>);  Bibliothek/Header:  Die Implementierung erfolgt über die folgende Header-Datei:  arraySort.h  Erklärung:  <arrayLength> = Länge des Arrays als Integer-Wert  &<array[0]> = Die Adresse des Arrays an Index 0. Hierzu einfach den Array an Index 0 aufrufen und ein „&“ davorsetzen. |
| Aufprallgeschwindigkeit nach einem freien Fall  Der Prototyp der Funktion lautet: void Funktion(double hoehe, double ergebnis).  Der Parameter double hoehe lässt den Nutzer entschieden von welcher Höhe, das jeweilige Objekt fallen gelassen wird. Dies ist entscheidend für die Rechnung der Aufprallgeschwindigkeit. Zurückgegeben wird die Aufprallgeschwindigkeit aus dem freien Fall, von der ausgesuchten Höhe des Nutzers. Der Parameter double ergebnis speicher speichert das Ergebnis der Rechnung.  Um auch die Funktion nutzen zu können, muss die funktion.h auch Header genannt in die Bibliothek mit eingebunden werden. |
| Volumens eines Quaders   |  |  | | --- | --- | | Prototyp | int VolumenQuader (int Laenge, int Hoehe, int Breite); | | Parameter | Int Laenge des Quaders  Int Hoehe des Quaders  Int Breite des Quaders | | Rückgabe | Das Volumen eines Quaders errechnen | |
| Oberflaeche eines Quaders  Name: quaderflaeche  Prototyp: int quaderflaeche (int laenge, int breite, int hoehe)  Parameter: int laenge, int breite, int hoehe  Rückgabe: Flächeninhalt eines Quaders  Bibliothek: funktion.h |
| Volumens einer Kugel |
| Oberflaeche einer Kugel  Der Prototyp der Funktion ist „void oberflaecheberechnung(double Radius);“.  Der Parameter „double Radius“ ist der Radius der von einem beliebigen Kreis eingegeben werden muss für die Funktion. Der Rückgabetyp ist die Oberfläche eines Kreises von dem der Radius angegeben wurde.  Um die Funktion nutzen zu können muss man die Bibliothek "Hbfi.2017.h" einfügen. |
| Volumens einer Pyramide  Die Funktion berechnet mit Hilfe von 3 Werten (Länge, Breite und Höhe) den Volumen einer Pyramide. Die Werte müssen eingefügt werden. Die Funktion gibt das Ergebnis am Ende aus nicht eigenständig aus.  Fügen Sie die Werte hinzu und geben sie das Ergebnis aus. |
| Oberflaeche einer Pyramide   |  | | --- | | Prototyp:  Int Pyramiden\_Funktion( int hoehe, int breite,int leange)  Parameter:  Int hoehe = Hoehe der Pyramide  Int leange = Leange der Grundkante  Int breite = Breite der Pyramide  Rückgabetyp:  Die Funktion rechnet zuerst den Flächeninhalt aus und danach die Mantelfläche. Dies wird zusammen addiert und das ist dann der Flächeninhalt in cm².  Bibliothek:  Pyramiden\_Funktion.h | | Oberflächenberechnung einer Quadratischen Pyramide  W-Fragen  Warum? Das Programm wurde erstellt falls der Kunde vorhat eine Pyramide zubauen und wissen möchte wie viel Oberfläche er zu behandeln hat.  Wie? Mit Hilfe zwei einfacher fragen kann die Antwort ermittelt werden.  Wer? Der Kunde kann das Programm eigenständig bedienen.  Die main.cpp  Anfangs wird die pyra.h Datei beigefügt. „grund, hoehe, ergebnis“ werden als int variablen festgelegt. Jetzt werden zwei fragengestellt. Es wird nach der Grundkante und nach der Seitenhöhe gefragt. Diese antworten werden in die Variablen gepackt. Jetzt wird die Funktion ausgeführt.  Die pyra.h  Hier wird die Funktion als void angegeben und die gebrauchten Variablen.  Die pyra.cpp  Anfangs wird die pyra.h Datei beigefügt. „grund, hoehe, ergebnis“ werden auch hier als int variablen festgelegt. Jetzt wird der Funktion ihre Funktion gegeben. Dort ist die Rechnung und die Ausgabe drin. Ist die Funktion ausgeführt wird das Programm weiter in der main gespielt. | |
| Volumens eines Zylinder  Prototyp:  double zylinder\_volumen(double hoehe, double radius)  Parameter:  double hoehe = Hoehe des Zylinders  double radius = Radius des Zylinders  Rückgabewert:  Diese Funktion rechnet das Volumen eines Zylinders aus und Gibt dies dann als Wert zurück. |
| Oberflaeche eines Zylinder |
| Momentanwerts zweier überlagerter Sinusförmiger Spannungen |
| Zahl aus dem Dezimal- in das Binaersystem  Prototyp: int convert(int i)  Parameter: int i: Dezimalzahl  Rückgabewert: Binärzahl von i  Bibliothek/Header: hbf2I\_2017.h |
| Berechnung des aktuellen Wertes eines Anuuitaetendarlehens |
| Nullstellen eines Polynoms zweiten Grades  Prototype: void Nullstellenberechnung (double a0, double a1, double a2)  Parameter: double a0 = x² Wert, double a1 = x Wert, double a2 = Y-Achsenabschnitt  Rückgabe: Die Funktion gibt nix zurück da es die Werte gleich ausgibt  Bibliothek: Man benötigt die cmath-Libary. |