|  |  |  |
| --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)***  C:\Users\norsu\Desktop\jata negara.png | | |
| **KOD DAN NAMA PROGRAM NOSS** | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP NOSS** | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | **CU05/WA3 - DEBUG APPLICATION CODE** | |
| **NAMA PROGRAM KV** | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | KPD2033 FUNDAMENTAL OF PROGRAMMING | |
| **NO DAN TAJUK STANDARD KANDUNGAN KSKV** | K3 WRITE MODULE CODE | |
| **NO. KOD NOSS** | IT-010-3:2016-C05 / P(23/24) | Muka Surat : 1 Drp : 13 |
| **NO. KOD KSKV** | KPD2033/P(9/13) |

**TAJUK/***TITLE***:**

**PROGRAMMING LANGUAGE MODEL**

**TUJUAN/***PURPOSE* **:**

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

* Jenis Bahasa pengaturcaraan yang digunakan.
* Model-model Bahasa pengaturcaraan

i.Structured Programming

ii.Procedural Programming

iii.Object Oriented Programming(OOP)

**PENERANGAN/***INFORMATION* **:**

Jenis Bahasa Pengaturcaraan yang digunakan adalah**.**

* **Java**

J**ava adalah bahasa berasaskan objek** yang dicipta pada tahun 1995 oleh James Gosling. Ia mempunyai ciri-ciri C++ dan Simula.

Java menukar aturcara Java kepada byte code, bukan kepada kod objek, dan byte code ini boleh dilarikan pada mana-mana sistem pengendalian dengan menggunakan mesin maya Java tanpa sebarang perubahan.

Kaedah ini membolehkan satu perisian Java bersifat mudahalih antara banyak sistem komputer.

Pada mulanya, Java banyak digunakan dalam menghasilkan grafik dan animasi pada laman Internet. Pakar menjangkakan yang Java akan menjadi bahasa pengaturcaraan yang paling meluas penggunaannya di masa hadapan. Java telah popular dengan slogan, Tulis Sekali, Dilarikan di mana-mana, (WRite Once, Run EveryWhere).

Contoh aturcara Java untuk aplikasi tiket wayang seperti di dalam Rajah 1.

|  |
| --- |
|  |

Rajah 1{ Keratan antaramuka JAVA

Terdapat tiga jenis model pengaturcaraan iaitu:

i.**Structured programming(Pengaturcaraan berstruktur)**

ii. **Procedural programming(bahasa pengaturcaraan bertatacara)**

**iii.Bahasa Berorientasikan Objek(OOP)**

i.**Structured programming(Pengaturcaraan berstruktur)**

**Adalah satu subset kepada pengaturcaraan prosedural yang membentuk struktur logik** pada program yang sedang ditulis bagi menjadikan ia lebih efisien dan lebih mudah untuk difahami dan diubah suai. Guna objek .Rekabentuk atas bawah mengabungkan data manakala fungsinya kesemua program dipecahkan kepadaadalah untuk merekacipta objek bahagian - bahagian kecil dan dikenali sebagai modulsetiap objek mempunyai hubungan Selalu digunakan untuk mengurus antara satu sama lain.

Cth: C++, Smalltalk, Java dan Objective-C, Object-Oriented, Pascal, Delphi, Beta, HyperTalk, Limbo, Python dan Eifel.

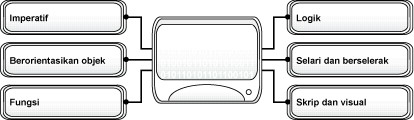
ii. **Procedural programming(bahasa pengaturcaraan bertatacara)**

**Ia dikenali juga sebagai bahasa bertatacara. Bahasa Bertatacara bermaksud program terdiri dari siri-siri arahan yang disusun mengikut tertib yang akan digunakan untuk menyelesaikan masalah.**Cth: FORTRAN, C, ADA, COBOL, Simula, APL, BCPL, BASIC, PL/1, Pascal, Logo, Modula, dan sebagainya.

Ciri-ciri

Bahasa pengaturcaraan boleh dikelaskan berdasarkan ciri-ciri bahasa tersebut seperti imperatif, berorientasikan objek, fungsi, logik, selari, berselerak dan skrip dan visual seperti yang ditunjukkan dalam Rajah 1.2. Setiap ciri-ciri ini mempunyai kebaikan dan kelemahan. Bahagian seterusnya membincangkan kebaikan dan kekurangan setiap ciri di atas.

Tajah 2:*Ciri-ciri bahasa pengaturcaraan*



* **Bahasa Imperatif(Berstruktur)**

**Maksud imperatif adalah arahan, suruhan atau perintah.**

Dengan itu, bahasa imperatif merupakan bahasa di mana setiap arahan untuk menyelesaikan masalah perlu dinyatakan dengan jelas. Setiap arahan untuk menyelesaikan masalah akan dilaksanakan langkah demi langkah. Bahasa imperatif boleh diibaratkan seperti satu resepi masakan. Dalam bahasa ini, kaedah yang digunakan adalah ‘buat A, kemudian buat B’. Contoh-contoh bahasa imperatif adalah seperti yang ditujukkan di dalam Jadual 6.3. Ada masanya, ia juga dikenali sebagai bahasa prosedur. Idea asas dalam bahasa imperatif adalah, pengguna perlu menyatakan setiap langkah yang diperlukan dalam menyelesaikan masalah. Jadual

|  |  |
| --- | --- |
| Bahasa Imperatif | |
| C | Algol 68 |
| Ada | PL/1 |
| C++ | Simula |
| Pascal | Turing |
| Modula 2 | Oberon |
| FORTRAN | BASIC |
| COBOL |  |

Rajah 3: Contoh-contoh Bahasa Imperatif

Sebagai contoh, kita diminta mencetak semua nombor ganjil antara 1 dan 40, maka kod dalam bahasa imperatif adalah seperti berikut:-

for(i=1;i<=40;i++) if (i % 2 != 0) printf(“ganjil %d\n”,i);

**iii.Bahasa Berorientasikan Objek**

**Bahasa berorientasikan objek dicipta untuk membangunkan sistem yang besar dan kompleks.Keperluan untuk bahasa berorientasikan objek timbul apabila bahasa yang sedia ada** tidak lagi dapat memenuhi keperluan sistem yang semakin kompleks. Kebaikan berorientasikan objek adalah kemudahan dalam pengurusan sistem, kemampuan untuk mengubah sistem secara teratur dan penggunaan semula komponen dalam sistem yang baru. Struktur bahasa berorientasikan objek membuatkan sesuatu program lebih mudah untuk direka bentuk dan difahami. Bahasa berorientasikan objek mengendalikan imej, audio, video dan bunyi lebih mudah dari bahasa prosedur (generasi ketiga).Idea asas ialah untuk memfokus kepada objek dalam sesuatu sistem, bukan kepada fungsi yang berlaku dalam satu sistem.

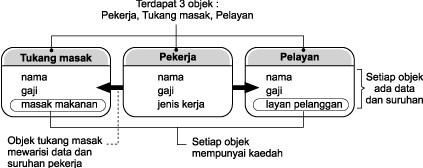
Konsep penting dalam bahasa berorientasikan objek adalah kelas, objek, enkapsulasi, warisan, polimorphisma dan ikatan dinamik.

Satu kelas menerangkan tentang satu objek, di mana objek ini mengandungi data tentang objek dan suruhan yang boleh diproses oleh objek. Sebagai contoh, pengguna ingin memodelkan satu restoran seperti yang ditunjukkan dalam Rajah 1.3. Kaedah seperti di dalam Rajah 1.4 boleh digunakan dalam bahasa berasaskan objek.

***Rajah 3***

*:*

*Objek dalam pengaturcaraan berorientasikan objek*



|  |
| --- |
| class Pekerja: def \_\_init\_\_(self,nama,gaji=0):  self.nama = nama self.gaji = gaji def kenaikanGaji(self,peratus):  self.gaji = self.gaji + (self.salary\*peratus)  def kerja(self):  print self.nama, ‘buat kerja’ def \_\_repr\_\_(self):  return ‘Pekerja: name:%s gaji=%s’ %  (self.nama , self.gaji)  class TukangMasak(Pekerja): def \_\_init\_\_(self,name):  Pekerja.\_\_init\_\_(self,nama,500) def kerja(self): print self.nama,’masak lauk’  class Pelayan(Pekerja)  def \_\_init\_\_(self,nama)  Pekerja.\_\_init\_\_(self,nama,400) def kerja(self) print self.name,’melayan pelanggan’ |

Rajah 4: Keratan aturcara bahasa berasaskan objek

Kalau kita teliti, kod ini menerangkan tentang tiga objek; Pekerja, TukangMasak dan Pelayan. Objek Pekerja mempunyai data tentang nama, gaji dan suruhan bagaimana mengira kenaikan gaji. Objek TukangMasak dan Pelayan mempunyai data tentang kerja yang dilakukan seperti “masak lauk” dan “melayan pelanggan”.

Konsep warisan membolehkan pengguna menghasilkan satu objek baru berasaskan objek yang sedia ada.

Sebagai contoh, objek TukangMasak adalah warisan dari objek Pekerja. Ini bermakna, objek TukangMasak mempunyai data dan suruhan yang sama dengan objek Pekerja. Perhatikan bahawa objek TukangMasak menakrifkan semula maklumat iaitu;

def kerja(self): print self.nama, ‘buat kerja’

kepada

def kerja(self): print self.nama,’masak lauk’

Kaedah di mana satu objek mempunyai data dan suruhan disebut enkapsulasi. Enkapsulasi bermaksud memasukkan ke dalam bekas. Polimorphisma merujuk kepada banyak bentuk iaitu data dalam objek boleh terdiri dari berbagai bentuk. Sebagai contoh, data self.gaji tidak dinyatakan bentuknya, sama ada integer, titik terapung atau aksara.

Penghantaran mesej digunakan apabila satu objek ingin berhubung dengan satu objek yang lain. Interaksi antara objek adalah melalui mesej. Konsep pengaturcaraan berasaskan objek agak sukar diterima pada mulanya, tetapi penggunaan konsep ini boleh digunakan untuk membangunkan sistem yang besar dan kompleks. Secara ringkas, tiga konsep asas dalam bahasa berasaskan objek adalah:

1. **Enkapsulasi:** bermaksud objek mempunyai data dan suruhan.
2. **Warisan**: satu objek boleh mewarisi data dan suruhan dari objek lain secara hirarki. Kelas adalah gabungan objek yang dijana dan mempunyai ciri yang sama.
3. **Polimorphisma**: bermaksud “banyak bentuk”. Data dalam objek tidak dinyatakan bentuknya. Poli, dari perkataan asal poly, bermaksud banyak sementara morphisma, dari perkataan morph, bermaksud berubah bentuk.

Satu kebaikan bahasa berasaskan objek ialah guna semula. Sesuatu objek yang telah diisytihar boleh diguna semula dalam aturcara yang lain. Ini memudahkan dan mempercepatkan pembangunan aplikasi. Bahasa berasaskan objek mula diperkenalkan pada tahun 1970 dalam bahasa Simula, diikuti oleh bahasa SmallTalk dan C++. Contoh-contoh bahasa objek adalah C++, Smalltalk, Java dan Objective-C, Object-Oriented, Pascal, Delphi, Beta, HyperTalk, Limbo, Python dan Eifel. Bahasa jenis ini mulai mendapat perhatian besar pada dekad 80an dan 90an.

* **Bahasa Berasaskan Logik**

Bahasa berasaskan logik lebih memfokus kepada penyataan masalah, bukan bagaimana untuk menyelesaikan masalah.

Sistem yang bertanggungjawab untuk menyelesaikan masalah yang dinyatakan. Bahasa ini berasaskan logik, iaitu berasaskan fakta dan peraturan. Fakta dan peraturan dinyatakan dalam bentuk penyataan if-then. Sebagai contoh, pengguna boleh menyatakan fakta dan peraturan seperti berikut:

if X ayah Y dan Y ayah Z, then X adalah datuk Z.

Dalam menulis aturcara logik, pengguna menggunakan logik Horn, iaitu satu penyataan yang menggunakan if-then.

Semua fakta dan peraturan akan dinyatakan dalam bentuk logik Horn. Selanjutnya, pengguna akan menyatakan masalah yang ingin diselesaikan. Sistem akan cuba menyelesaikan masalah tersebut berasaskan fakta dan peraturan yang telah diberikan. Sebagai contoh, kita boleh menghasilkan maklumat berikut berasaskan logik Horn:

gembira(x)<-kaya(x),terkenal(x) gembira(x)<-muda(x),dalam\_percintaan(x) gembira(x)<-tua(x),bijaksana(x) terkenal(x)<-pelakon(x) terkenal(x)<-penyanyi(x)

Seterusnya kita nyatakan fakta berikut:

pelakon{ali} kaya(ali)

Akhir sekali, kita boleh menanyakan kepada sistem soalan berikut, yang bermaksud adakah ali gembira?:

?-gembira(ali)

Sistem akan memberikan jawaban ya atau tidak, berdasarkan maklumat yang diberikan sebelum ini. Contoh bahasa logik adalah Prolog.

Beberapa contoh bahasa yang dibincangkan di atas dan bidang kegunaannya untuk bahasa ditunjukkan dalan Jadual 1.4.

|  |  |  |
| --- | --- | --- |
| Bahasa Pengaturcaraan | Bidang | Tahun |
| FORTRAN | Penghitungan saintifik. | 1956 |
| Lisp | Penghitungan simbol, kecerdasan buatan. | 1957 |
| C | Pembangunan sistem. | 1974 |
| COBOL | Perniagaan. | 1960 |
| Visual Basic | Pembangunan sistem, GUI | 1990 |
| C++ | Pembangunan sistem besar dan kompleks. | 1984 |
| Java | Internet, GUI, pembangunan sistem. | 1995 |
| Perl | Internet, CGI, teks | 1990 |

Jadual 1: Contoh-contoh Bahasa Pengaturcaraan dan Bidang Kegunaannya

* Bahasa Berasaskan Fungsi

Bahasa berasaskan fungsi menekankan kepada apa yang perlu dihitung, tidak kepada bagaimana untuk menghitung. Aturcara adalah dalam bentuk fungsi, iaitu satu fungsi yang hampir setara dengan fungsi matematik. Bahasa fungsi berdasarkan kepada penilaian terhadap ungkapan yang mengandungi fungsi. Kod yang ditulis dalam bahasa fungsi lazimnya singkat dan padat. Sebagai contoh, untuk mendapat nombor ganjil antara 1 dan 20, aturcara berasaskan fungsi adalah seperti berikut:

[x|x<-[1..20];x mod 2 = 1]

Contoh bahasa fungsi adalah seperti Lisp, FP, Scheme, Common Lisp, Hope, Standard ML, Miranda, Haskell, Hugs, Lucid, FP.

* Lain-lain

Bahasa skrip adalah bahasa terjemahan yang lazimnya dihasilkan untuk satu tujuan yang tertentu. Ia boleh digunakan untuk membangunkan prototaip secara cepat dan pantas. Bahasa ini bersifat mudahalih antara berbagai jenis pelantar komputer. Sistem yang dibangunkan dalam bahasa skrip adalah lebih perlahan. Contoh bahasa skrip adalah awk, sed, Perl, Python, REXX, rebol dan ruby.

Selain dari ciri-ciri yang disebut, terdapat juga bahasa-bahasa lain seperti:

1. Constraint programming
2. Accessed-oriented programming
3. Single data structure programming sebagai contoh, bahasa APL menganggap semua data dalam bentuk matrik atau tatasusunan, bahasa Icon menganggap semua data dalam bentuk ungkapan.
4. Bahasa pangkalan data. Contoh: SQL.

**CONTOH-CONTOH ATURCARA**

Bahasa C

Ia merupakan satu bahasa yang kecil dengan ciri-ciri yang menarik. Bahasa C banyak digunakan dalam pembangunan sistem dan aplikasi.

Aturcara dalam C adalah mudahalih, di mana pengguna hanya perlu kompil semula aturcara tersebut pada sistem pengendalian yang berlainan untuk menghasilkan kod objek yang sesuai.

C merupakan bahasa yang paling banyak digunakan untuk pembangunan sistem dan aplikasi. Sebagai contoh, sistem pengendalian Linux dan Unix dibangunkan dalam bahasa C. Kursus C merupakan satu kursus teras dalam pembelajaran komputer sains di mana-mana universiti. Bahasa C agak sukar dipelajari oleh pengguna baru tetapi ia merupakan satu bahasa yang serba boleh dan amat berkuasa.

Rajah 5 adalah contoh aturcara C untuk mendapat nombor jati berasaskan algoritma eratosthenes.

/\* sieve of eratosthenes \*/

|  |
| --- |
| #DEFINE N 10000 main ()  { int i, j, a[N]; for (i = 2; i < N; i++) a[i] = 1; for (i = 2; i < N; i++) if (a[i]) for (j = i; j < N / i; j++) a[i \* j] = 0; for (i = 2; i < N; i++) if (a[i]) printf (‘ ‘); printf (‘\\n’);  } |

Rajah 5: Keratan aturcara C

**C++**

C++ adalah bahasa berasaskan objek yang dicipta di Bell Lab oleh Bjarne Strousrup (http://www.research.att.com/~bs/homepage.html) pada tahun 1983.

Ia menambahkan ciri-ciri objek kepada bahasa C berasaskan idea dari bahasa Simula. Pada mulanya ia dikenali sebagai bahasa C with Classes. Bahasa ini banyak digunakan dalam pembangunan sistem yang besar dan kompleks seperti pensuisan telefon, sistem GUI, pengawasan trafik jalanraya dan lain-lain.

Rajah 1.11 menunjukkan contoh aturcara berasaskan objek dalam C++;

|  |
| --- |
| void gen\_array::read(istream& in, string s)  { clear(); cout << s; int i = 0; while (in && i < sz) read\_el(v[i++],in); while (i < sz) init\_entry(v[i++]);  }  gen\_array::gen\_array(int a, int b)  { if (a>b) error\_handler(1,string(“array: bad  [%d..%d]”,a,b));  Low = a; High = b; sz = b-a+1; v = new GenPtr[sz]; if (v==0) error\_handler(99,”array: out of memory”); last = v+sz-1;  } |

Rajah 6: Keratan aturcara C++

Perl

Perl adalah bahasa skrip yang boleh digunakan dalam memproses teks, CGI, menghubung pangkalan data dan sebagainya. PERL bersifat mudah alih.Contoh aturcara perl untuk memproses dokumen adalah seperti di dalam Rajah 8.

|  |
| --- |
| @in = split(/[&;]/,$in); push(@in, @ARGV) if $cmdflag;  foreach $i (0 .. $#in) { $in[$i] =~ s/\+/ /g; ($key,$val) = split(/=/,$in[$i],2);#splits on the first =.  $key =~ s/%([A-Fa-f0-9]{2})/pack(“c”,hex($1))/ge;  $val =~ s/%([A-Fa-f0-9]{2})/pack(“c”,hex($1))/ge;  $in{$key} .= “\0” if (defined($in{$key}));  $in{$key} .= $val;  }  $name=~/([-\w]+)/;#This allows $insfn{$name} to be untainted if (defined $1) {  $insfn{$1} .= “\0” if defined $insfn{$1};  $insfn{$1} .= $fn if $fn;  }$ |

Rajah 8: Keratan antaramuka Perl

**Python**

Python adalah satu bahasa skrip berasaskan objek yang boleh digunakan untuk pembangunan prototaip, pemprosesan data berasaskan teks, menghubungi pangkalan data dan sebagainya.

Antara aplikasi yang telah dibina menggunakan python adalah peluncur web, buku alamat, pengurusan rangkaian komputer dan perisian astronomi. Python merupakan satu alternatif kepada Perl.

Contoh aturcara python di dalam Rajah 9 boleh digunakan untuk menghitung bilangan perkataan dalam satu dokumen;

|  |
| --- |
| from sys import \* from string import \* count = {} for line in open(argv[1], ‘r’).readlines(): for word in split(line): if count.has\_key(word):  count[word] = count[word] + 1 else:  count[word] = 1 words = count.keys() words.sort() for word in words:  print “%15s\t%10d” % (word, count[word]) |

Rajah 9: Keratan aturcara Python

**SOALAN/***QUESTION* **:**

1. Senaraikan tiga model bahasa pengaturcaraan.
2. Terangkan Bahasa berasaskan logik.
3. Berikan contoh bagi pengaturcaraan berstruktur.

**RUJUKAN/***REFERENCE* **:**

1. Keith Cooper, Linda Torczon(2011), Engineering a Compiler (2nd Edition), Morgan Kaufmann, ISBN: 978-0120884780
2. Todd Zaki Warfel (2009), Prototyping: A Practitioner's Guide, Louis Rosenfeld Media, LLC, ISBN I-933820-21-7
3. Steve McConnell (2004), Code Complete: A Practical Handbook of Software Construction, (2nd Edition),Microsoft Press, ISBN: 978-0-7356-1967-8
4. Michael L. Scot (2009) , Programming Language Pragmatics, Third Edition, Morgan Kaufmann, ISBN-13: 978-0123745149
5. Simon Marlow (2013), Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming (1st Edition), O'Reilly Media, ISBN-13: 978-1449335946