3. Dodatek

3.1. Pliki naglówkowe

Pliki naglówkowe powinny zawierac:

» stale jawne **#define** TRUE 1

» funkcje makra **#define** MAX(x, y) ((x) > (y) ? (x) : (y))

» dyrektywy **#include** <iostream.h>

» prototypy funkcji np. **void** wyswietl_float(**float**);

» deklaracje definicyjne typów (**struct, union, enum**) np.

```
struct punkt
  { float x, y;
    public:    punkt (float, float);
               void przesun (float, float);
  };
```

» definicje stalych **const int** max = 3;

» definicje funkcji typu **inline**

```
inline int Wiekszy(int x, int y) {return x > y; }
```

» deklaracje nazw **struct** kolo;

» deklaracje zmiennych **extern int** zmienna;

» wyliczenia **enum** Boolean {False, True};

» szablony funkcji i klas **template** <class T>

```
class stos { // .....}
```

Uwaga:

Nie nalezy nigdy wstawiac do pliku naglówkowego:

1. definicji zwyklych funkcji: **int Wiekszy(int** x, **int** y) {**return** x > y; }
2. definicji zmiennych: **int zmienna;**
3. definicji stalych agregatów (tablica, obiekt bez konstruktorów, skladowych prywatnych i chronionych, klas podstawowych i funkcji wirtualnych):

```
const char tab[] =”aaa”;
```

3.2. Programy wieloplikowe

Dyrektywy preprocesora i kompilacja wieloplikowa (projekty)

Polecenia dla preprocesora:

1) Dyrektywa **#include** <stdio.h> oznacza dolaczenie w miejscu wystapienia polecenia standardowego pliku naglówkowego z deklaracjami typów, funkcji itp, natomiast **#include** ”tablica1.h” dolaczenie pliku naglówkowego uzytkownika

2) Klauzula **#define nazwa** oznacza makrodefinicje, **#undef nazwa** uniewaznia makrodefinicje

3) Polecenia kompilacji warunkowej pozwalaja na kompilacje tylko jednej z sekcji instrukcji:

#if *wyrazenie1*
sekcja _instrukcji1
#elif *wyrazenie2*
sekcja _instrukcji2
....
#else
koncowa_sekcja_instrukcji
#endif

4) Polecenia warunkowej kompilacji uniemozliwiaja wielokrotnego dolaczania tego samego pliku naglówkowego, lub jego fragmentu podczas kompilacji wieloplikowej:

#ifndef nazwa
#define nazwa
deklaracje *//deklaracje czyta kompilator tylko raz, podczas definiowania makro **nazwa***
#endif,

#ifdef nazwa
deklaracje *//kompilator czyta te deklaracje, gdy zdefiniowano makro **nazwa***
#endif

Tworzenie projektu umozliwiajacego kompilacje oraz laczenia plików

1. Nalezy uruchomic **Borland C++** i wybrac opcje **Open Project** z menu **Project**. Nalezy wpisac nazwe pliku projektowego w polu **Open Project File** z rozszerzeniem **PRJ**, np. **main.prj**. Nazwa nadana plikowi **PRJ** bedzie nadana programowi wynikowemu **EXE** czyli **main.exe**. Po nadaniu nazwy nalezy nacisnac klawisz **ENTER.** U dolu ekranu pojawi sie okno o nazwie **Project : main**

2. Aby dodac pliki do projektu nalezy nacisnac klawisz **INSERT** lub wybrac opcje **Add Item** z menu **Project**. Pojawi sie okno o nazwie **Add to Project List.** W polu **Name** nalezy wpisac nazwy plików (przez wybór z listy), które nalezy dodac do projektu (w przykladzie pliki **tabwsk.cpp**, **dodatki.cpp, we_wy.cpp** oraz **main.cpp**). Nie nalezy dodawac plików naglówkowych. Po zakonczeniu nalezy nacisnac przycisk **Done** w oknie **Add to Project List.**

3. Aby skompilowac i polaczyc pliki zródlowe podane w projekcie, nalezy wybrac opcje **Make** z menu **Compile** lub nacisnac klawisz **F9.** Nastapi kompilacja i utworzenie plików **OBJ,** a nastepnie polaczenie tych plików i utworzenie programu wynikowego **main.exe**. Mozna równiez uruchomic program w srodowisku za pomoca polecenia **Run** z menu **Run** lub gotowego programu **main.exe** na poziomie systemu operacyjnego.