Contents

1 components.h **1**

2 components_dump.c **1**

3 comp_generator.c **2**

4 view_table.c **3**

5 cool_components.c **4**

6 Запрос SQL **5**

1 components.h

```
#ifndef __components_h__
#define __components_h__

typedef struct {
int id;
char surname[50];
int processorQuant;
char processorType[50];
int memoryWeight;
char controlerType[50];
int videoMemoryWeight;
char winchesterType[50];
int winchesterQuant;
int winchesterWeight;
int integControlerQuant;
int outerDevicesQuant;
char OC[50];
} component;

#endif
```

2 components_dump.c

```
#include <stdio.h>
#include <string.h>

#include "components.h"

void Usage()
{
printf("Usage: program filename\n");
}

int readComponent(component *comp)
{
return scanf("%s %d %s %d %s %d %s %d %d %d %d %s\n", &comp->surname, \
&comp->processorQuant, &comp->processorType, &comp->memoryWeight, &comp->controlerType, \
&comp->videoMemoryWeight, &comp->winchesterType, &comp->winchesterQuant,\
&comp->winchesterWeight, &comp->integControlerQuant, &comp->outerDevicesQuant, &comp->OC) == 12;
}
```

```
int main(int argc, char * argv[])
{
if(argc != 2)
{
Usage();
return 1;
}
component comp;
FILE *out = fopen(argv[1], "w");
if(!out)
{
perror("Can't open the file");
return 2;
}
int id = 1;
while(readComponent(&comp))
{
comp.id = id;
fwrite(&comp, sizeof(comp), 1, out);
id++;
}
return 0;
}
```

3 comp_generator.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "components.h"

void Usage()
{
printf("Usage: program filename, components quantity\n");
}
const char names[10][50] = {"Timur", "Valera", "Masha", "Misha", "Anna", "Tanya", "Greror", "Max", "Vit
const char procType[3][50] = {"Intel", "AMD", "Pentium"};
const char contrType[3][50] = {"Keyboard", "Touchpad", "VoiceControler"};
const char winchType[2][10] = {"SSD", "HDD"};
const char OC[4][15] = {"Windows", "MacOC", "Linux/Ubuntu", "MS/DOS"};

void generateComponent(component *comp, int id)
{
comp->id = id;
strcpy(comp->surname, names[rand()%10]);
comp->processorQuant = 1+rand()%3;
strcpy(comp->processorType, procType[rand()%3]);
comp->memoryWeight = 1+rand()%1024;
strcpy(comp->controlerType, contrType[rand()%3]);
comp->videoMemoryWeight = 1+rand()%512;
strcpy(comp->winchesterType, winchType[rand()%2]);
comp->winchesterQuant = 1+rand()%4;
comp->winchesterWeight = 1+rand()%1024;
```

```
comp->integControlerQuant = 1+rand()%3;
comp->outerDevicesQuant = 1+rand()%3;
strcpy(comp->OC, OC[rand()%4]);
}

int main(int argc, char * argv[])
{
if(argc != 3)
{
Usage();
return 1;
}
component comp;
FILE *out = fopen(argv[1], "wb");
int n = atoi(argv[2]);
if(!out)
{
perror("Can't open the file");
return 2;
}
srand(n);
for(int i = 0; i < n; i++)
{
generateComponent(&comp, i+1);
fwrite(&comp, sizeof(comp), 1, out);
}
return 0;
}
```

4 view_table.c

```
#include <stdio.h>
#include <string.h>

#include "components.h"

void Usage()
{
printf("Usage: program filename\n");
}

void writeComponent(component *comp)
{
printf("%d\t%s\t%d\t%s\t%d\t%s\t%d\t%s\t%d\t%d\t%d\t%d\t%d\t%s\n", comp->id, comp->surname, \
comp->processorQuant, comp->processorType, comp->memoryWeight, comp->controlerType, \
comp->videoMemoryWeight, comp->winchesterType, comp->winchesterQuant,\
comp->winchesterWeight, comp->integControlerQuant, comp->outerDevicesQuant, comp->OC);
}

int main(int argc, char * argv[])
{
if(argc != 2)
{
Usage();
return 1;
}
```

```
component comp;
FILE *in = fopen(argv[1], "rb");
if(!in)
{
perror("Can't open the file");
return 2;
}
while(fread(&comp, sizeof(comp), 1, in)==1)
{
writeComponent(&comp);
}
return 0;
}
```

5 cool_components.c

```
    #include <stdio.h>
#include <stdlib.h>

#include "components.h"

void Usage()
{
printf("Usage: program filename, p\n");
}

/*
Чекаем всех студентиков, имена тех, у кого 2 процессора и внешних устройств <= p - выводим в ст. вывод
параметр p берем из параметров вызова, он должен идти после имени файла с данными студентиков
*/

int main(int argc, char * argv[])
{
if(argc != 3)
{
Usage();
return 1;
}
FILE * in = fopen(argv[1], "rb");
if(!in)
{
perror("Can't open file");
return 2;
}

int p = atoi(argv[2]);
component comp;
while(fread(&comp, sizeof(comp), 1, in) == 1)
{
if(comp.processorQuant == 2 && comp.outerDevicesQuant <= p)
{
printf("%d\t%s\n", comp.id, comp.surname);
}
}
return 0;
}
```

6 Запрос SQL

Описание таблицы: CREATE TABLE COMP (SURNAME CHAR[50], PROCQUANT INTEGER, PROCTYPE CHAR[50], MEMWEIGHT
Запрос: SELECT SURNAME FROM COMP WHERE(PROCQUANT=2 AND OUTERDEVQUANT <= p);