Zvýšení efektivity kódu

➢ Cykly

- Snížení výpadků stránek ve virtuální paměti.

  U jazyků C, C++ a dalších jsou dvourozměrná pole ukládána po řádcích.
  V následujících cyklech jsou prvky pole čteny po sloupcích, což může vést k vyššímu nároku na současnou přítomnost stránek v obsahem polí v paměti a tím i k větší pravděpodobnosti výpadku stránek.

```
for (sloupec=0; sloupec<POCETSLOUPCU; ++sloupec) {
  for (radek=0; radek<POCETRADKU; ++radek) {
    matice[radek][sloupec] = hodnota;
  }
}
```

  Cykly přepíšeme, aby pole bylo čteno po řádcích, tedy v pořadí, jak jsou jeho prvky uloženy v paměti, čímž stačí v paměti vždy jen jedna stránka.

```
for (radek=0; radek<POCETRADKU; ++radek) {
  for (sloupec=0; sloupec<POCETSLOUPCU; ++sloupec) {
    matice[radek][sloupec] = hodnota;
  }
}
```

- Cyklus s podmínkou – rozdělení na více cyklů bez podmínky.

```
for (int i=0; i<pocet; ++i) {
  if (operace == PRICTENI) {
    vysledek += hodnoty[i];
  }
  else {
    vysledek -= hodnoty[i];
  }
}
```

  Cyklu přepíšeme na dva, v kterých není vyhodnocování podmínky.

```
if (operace == PRICTENI) {
  for (int i=0; i<pocet; ++i) {
    vysledek += hodnoty[i];
  }
}
else {
  for (int i=0; i<pocet; ++i) {
    vysledek -= hodnoty[i];
  }
}
```

- Spojení cyklů, které mají stejný počet průchodů.

```
for (int i=0; i<pocet; ++i) {
  poleA[i] = hodnotaA;
}
for (int i=0; i<pocet; ++i) {
  poleB[i] = hodnotaB;
}
```

Místo dvou cyklů budeme mít jen jeden cyklus.

```
for (int i=0; i<pocet; ++i) {
  poleA[i] = hodnotaA;
  poleB[i] = hodnotaB;
}
```

- Rozvinutí cyklu.

```
for (i=0; i<pocet; ++i) {
  pole[i] = i;
}
```

Převedeme na cyklus s krokem 2, čímž se sníží počet průchodů cyklu přibližně na polovinu.

```
for (i=0; i<pocet-1; i+=2) {
  pole[i] = i;
  pole[i+1] = i+1;
}
```

Ošetříme situaci, kdy *pocet* je liché číslo.

```
if (i<pocet) {
  pole[pocet-1] = pocet-1;
}
```

Obdobně převedení na cyklus s krokem 3, čímž se sníží počet průchodů cyklu přibližně na třetinu.

```
for (i=0; i<pocet-2; i+=3) {
  pole[i] = i;
  pole[i+1] = i+1;
  pole[i+2] = i+2;
}
if (i<pocet) {
  pole[pocet-1] = pocet-1;
  if (i<počet-1) {
    pole[pocet-2] = pocet-2;
  }
}
```

- Zredukování těla cyklu – přesunutí částí, které se v cyklu nemění, před cyklus.

```
for (int i=0; i<pocet; ++i) {
  pole[i] = 2*hodnota + i;
}
```

Výpočet součinu dáme před cyklus (bude počítán je jedenkrát místo jeho výpočtu při každém průchodu cyklu).

```
int h=2*hodnota;
for (int i=0; i<pocet; ++i) {
  pole[i] = h + i;
}
```

- Záměna vnořených cyklů – vnější cyklus probíhá mnohem častěji než vnitřní cyklus.

```
for (radek=0; radek<100; ++radek) {
  for (sloupec=0; sloupec<5; ++sloupec) {
    matice[radek][sloupec] = hodnota;
  }
}
```

Počet vyhodnocení podmínky u vnějšího cyklu: 101
Počet vyhodnocení podmínky u vnitřního cyklu: 100*6
Celkem: 101+100*6=701

```
for (sloupec=0; sloupec<5; ++sloupec) {
  for (radek=0; radek<100; ++radek) {
    matice[radek][sloupec] = hodnota;
  }
}
```

Počet vyhodnocení podmínky u vnějšího cyklu: 6
Počet vyhodnocení podmínky u vnitřního cyklu: 5*101
Celkem: 6+5*101=507