

# プログラムの作成方法

# Linux環境

- リモートデスクトップでLinux環境を使用
  - 使用法は<https://imc.tut.ac.jp/wiki/Guide/>
  - ホームディレクトリはZドライブ
  - VPNを用いれば、自宅からも使用可能
- 
- ローカルな環境が必要なら
  - Windowsなら、cygwinにgcc(g++)をインストール
  - Macなら、Xcodeをインストール

# VSCode

- エディタはVSCodeがおすすめ
- 拡張機能
  - Japanese Language Packをインストール
  - C/C++をインストール
- 全角、半角スペースを表示
  - zenkakuをインストール、表示→コマンドパレットからEnable Zenkakuで有効化
  - ファイル→ユーザー設定→Render Whitespace=allに設定

# サンプルプログラム

- `svg.h`: SVGで図形を描くクラスを宣言
- `svg.cpp`: SVGで図形を描くクラスを実装
- `Makefile`: 自動コンパイルの設定ファイル
- これらはコピーして今後の課題で使用する
- `sample.cpp`: `svg`を使ったサンプルプログラム

# Makefile

- 手動でコンパイルすると大変
- `g++ -c sample.cpp`
- `g++ -c svg.cpp`
- `g++ -o sample sample.o svg.o`
- Makefileを用意してmakeを用いれば簡単
- makeで自動でコンパイル
- `make clean`で不要なファイルの削除

# Makefile

```
PROGRAM = sample
OBJS = sample.o svg.o
CXX = g++
CFLAGS = -Wall -O2 -std=c++11
LFLAGS = -Wall -lm
.PHONY: all
all: clean $(PROGRAM)
$(PROGRAM): $(OBJS)
    $(CXX) $(LFLAGS) -o $(PROGRAM) $^
.SUFFIXES: .cpp .o
.cpp.o:
    $(CXX) $(CFLAGS) -c $<
.PHONY: clean
clean:
    $(RM) $(PROGRAM) $(OBJS)
```

# 課題1のMakefile

```
PROGRAM = main
OBJS = main.o Circle.o
CXX = g++
CFLAGS = -Wall -O2 -std=c++11
LFLAGS = -Wall -lm
.PHONY: all
all: clean $(PROGRAM)
$(PROGRAM): $(OBJS)
    $(CXX) $(LFLAGS) -o $(PROGRAM) $^
.SUFFIXES: .cpp .o
.cpp.o:
    $(CXX) $(CFLAGS) -c $<
.PHONY: clean
clean:
    $(RM) $(PROGRAM) $(OBJS)
```

# 課題提出の注意事項

- google classroomを用いて提出してください。
- Makefileを用いてコンパイルできるようにしてください。
- ファイル一式をzipでまとめて、課題番号-学籍番号.zip(例えば1-B191234.zip)というファイル名にして提出してください。
- 演習室の環境で正常に実行される事を確認してから提出して下さい。（すなわち、演習室の環境で正常に動作する事が条件として課せられます。）